VOGUE: Try-On by StyleGAN Interpolation Optimization

Kathleen M Lewis[1,2*] kmlewis@google.com
Srivatsan Varadharajan[1] srivatsanv@google.com
Ira Kemelmacher-Shlizerman[1,3] kemelmi@google.com

[1]Google Research [2]MIT CSAIL [3]University of Washington

Figure 1: VOGUE is a StyleGAN interpolation optimization algorithm for photo-realistic try-on. Top: shirt try-on automatically synthesized by our method in two different examples. Bottom: pants try-on synthesized by our method. Note how our method preserves the identity of the person while allowing high detail garment try on.

Abstract

Given an image of a target person and an image of another person wearing a garment, we automatically generate the target person in the given garment. At the core of our method is a pose-conditioned StyleGAN2 latent space interpolation, which seamlessly combines the areas of interest from each image, i.e., body shape, hair, and skin color are derived from the target person, while the garment with its folds, material properties, and shape comes from the garment image. By automatically optimizing for interpolation coefficients per layer in the latent space, we can perform a seamless, yet true to source, merging of the garment and target person. Our algorithm allows for garments to deform according to the given body shape, while preserving pattern and material details. Experiments demonstrate state-of-the-art photo-realistic results at high resolution (512×512*).*

[1]Work done while the first author was an intern at Google Research.

1. Introduction

Online apparel shopping has become increasingly popular due to its convenience, and a large variety of products. Virtual try-on—the ability to computationally visualize a garment of interest on a person of one's choice—may amplify the shopping experience, as well as help reduce the environmental costs due to overproduction and returns.

A useful fashion try-on, however, requires high detail and high quality visualization, ideally indistinguishable from a photograph in a magazine. As a step towards this goal, we introduce a novel controllable image generation algorithm, named *VOGUE*, which seamlessly integrates person-specific components from one image with the garment shape and details from another image. Our experimental evaluation demonstrates state of the art photo-realistic results at the high resolution of 512×512 pixels.

We are motivated by the photo-realism and high resolution results of StyleGAN [20, 21] for faces, and use it as our starting point for fashion try-on. We first train a modified

StyleGAN2 network, conditioned on 2D human body pose, on 100K unpaired fashion photographs. (See discussion of paired vs. unpaired training data in section 2.) Once the model is trained, it remains fixed. Given a pair of images—a person image and a garment image— our method automatically finds the optimal interpolation coefficients per layer. Interpolation coefficients are applied on the latent representations of the two input images, and are used to generate a single output image where the person from the first input image is wearing the garment from the second image. Figure 1 demonstrates the results.

By optimizing for interpolation coefficients per layer, we are able to achieve semantically meaningful and high quality results. Unlike previous general GAN editing methods [6, 2], which require manual choice of noise injection structure or of clusters and fixed parameters for all layers, our method automatically computes the best interpolation coefficients by optimizing a loss function designed to preserve the identity and pose of the person while switching only the garment. Compared to try-on specific literature [22, 24, 30], our method outperformed the state of the art with respect to photo-realism and quality of the results. Extensive comparisons and ablation studies provide insight into why VOGUE outperforms others on the same task.

Our key contribution is an automatic interpolation optimization algorithm for fashion try-on from a single image, particularly pushing the limits of photo-realism and magazine-like quality for try on.

2. Related Work

Virtual Try-On (often abbreviated as VITON and VTON) has seen tremendous progress in the recent years. Given a pair of images (person, garment), the original VITON method [13] synthesizes a coarse try-on result, later refined, and warped with thin plate splines over shape context matching [3]. CP-VTON [30] adds geometric alignment to improve the details of the transferred garment. [17] incorporates adversarial loss in [30] to further improve image quality. PIVTONS [5] applies a similar concept on shoes rather than tops and shirts. [7] extends [13] to synthesize try on in various body poses. Further, GarmentGAN [26] separates shape and appearance to two generative adversarial networks. SieveNet [18] introduces a duelling triplet loss to refine details. ACGPN [32] aims to preserve the target person's identity in addition to the transferred clothes details, by accounting for semantic layout. SwapNet [27] first warps and then applies texture to transfer full outfits, rather than individual garments.

[22] and [33] incorporate learnings from StyleGAN [20] into try-on. ADGAN [22] conditioned the model on body pose, person identity, and multiple garments, where a separate latent code is generated to each of those components, and then combined into a single result by borrowing the needed parts from each image. This typically results in good transfer of uniform colors and textures but fails to synthesize the correct garment shape and texture details. [33] similarly conditions on pose and clothing items, but not for person's identity.

A key assumption of all above methods is availability of large *paired* training data, e.g., photographs of same person in various body poses wearing the same garment, or photograph of a person wearing a garment paired with separate garment images. Paired training data provides a ground-truth and a simpler design of losses. It is, however, a big limitation that tampers with quality and photo-realism of the results, since such paired data hard to obtain in large quantities required to train deep networks and particularly to generalize to the high variability of patterns, shapes, and details appearing in garments.

O-VITON [24] works with unpaired training data. It contains three stages: shape generation network, appearance generation network, both based on pix2pixHD [31], and an appearance refinement step. The shape and appearance generation network outputs are compared with the input image and segmentation in the loss function, the appearance refinement step is applied to each garment separately. The separation to three stages is what allows to work with unpaired data. Our algorithm, too, works with unpaired data, but with the key difference of doing all three stages in a *single* optimization within [21] architecture. By eliminating the need for three separate steps, as well as our StyleGAN2 conditioning, we enable higher photo-realism.

Related to our method are also conditional GAN networks [23, 4], and GAN editing methods [25, 2, 6, 15, 8, 1]. [2] uses a grid structure to inject noise into a GAN to achieve spatial disentanglement on a grid, and then edit the image. [6] further accounts for spatial semantics by K-means clustering the StyleGAN activation tensors, then searches for interpolation coefficients to do localized semantic editing. The latter is a baseline for our proposed algorithm. All those algorithms are not focused on apparel try-on, but mostly on face photos. Running those for try on does not produce good results as we show in the ablation part.

3. Method

In this section, we describe our VOGUE optimization algorithm for garment transfer. Given a pair of images generated by StyleGAN2, we show how to optimally interpolate between the generated images to accomplish try-on. We also describe how to use the network for any image, via projection of the image to our latent space, and then running VOGUE.

Problem Formulation Given an image I_p of a person p in some outfit, and an image I_g of a different person in a garment g, we aim to create a photo-realistic synthesis of

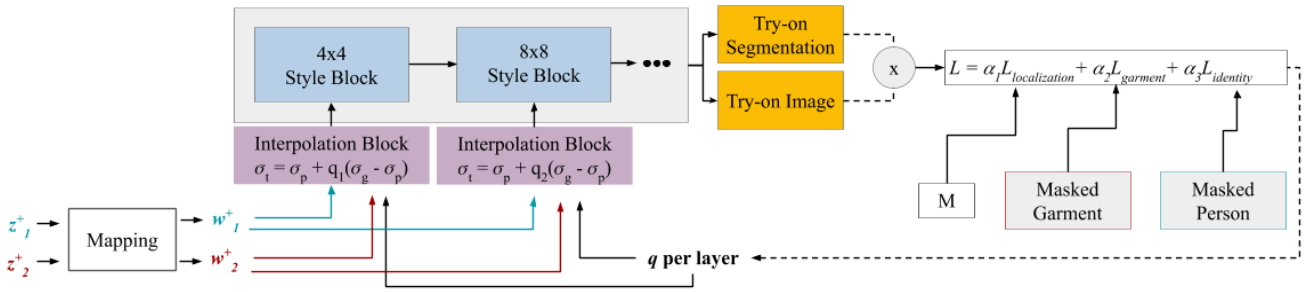

Figure 2: The try-on optimization setup illustrated here takes two latent codes z_1^+ and z_2^+ (representing two input images) and a pose heatmap as input into a pose-conditioned StyleGAN2 generator (gray). The generator produces the try-on image and its corresponding segmentation by interpolating between the latent codes using the interpolation-coefficients q. By minimizing the loss function over the space of interpolation coefficients, we are able to transfer garment(s) g from a garment image I_g, to the person image I_p.

Unclear: Segmentation motivation - maybe that the model hopefully learns to better disentangle the segments?

Also important since we don't have segmentation labels for actual images that we want to use this algorithm on

the person p in garment g.

Unclear: pose conditioned - I assume it's just that we feed in pose heatmaps instead of a constant thingy

The first step of our algorithm is to train a pose conditioned StyleGAN2, which can generate a photorealistic image of a person in some outfit given a 2D pose skeleton. We train our model to output RGB images as well as the garment and person segmentation in the image. Given a trained model, the second step is to optimize for interpolation coefficients at each layer to get the desired try-on result image I_t where person p will appear in garment g.

Important: At each layer

3.1. Pose-conditioned StyleGAN2

Generative Adversarial Networks (GANs) [12] have been shown to synthesize impressive images from latent codes. StyleGAN and StyleGAN2 [20, 21] in particular demonstrated state-of-the-art photo-realism on face images. The idea to combine progressive growing [19] and adaptive instance normalization (AdaIN) [16, 10, 9, 11] with a novel mapping network between the latent space, Z, and an intermediate latent space, W, encouraged disentanglement of the latent space. Transforming intermediate latent vector $w \in W$ into style vectors s further allowed different styles at different resolutions. Motivated by those advances we choose StyleGAN2 as our base architecture.

explanation of Z and W

Maybe s is our \sigma?

We train a StyleGAN2 model on fashion images, with two key modifications:

Important

- We replace the constant 4x4 block at the beginning of the generator with an encoder that takes as input pose keypoints represented as a heatmap. The keypoints are converted to a N_p channel heatmap representation where N_p is the number of pose keypoints (in our case, 17). Channels corresponding to keypoints that fall outside the cropped image are filled with zeros.
- We train our StyleGAN2 to output segmentations at each resolution/layer in addition to RGB images.

Figure 3 shows our modified StyleGAN2 architecture.

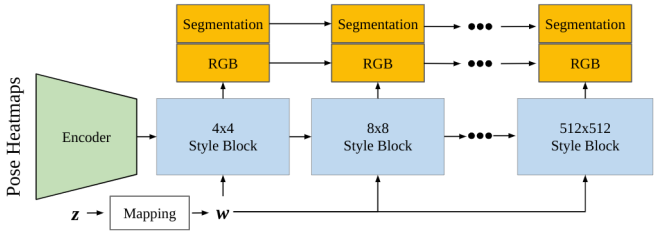

Figure 3: We trained a pose conditioned StyleGAN2 network, that outputs both an RGB image as well as a segmentation of the image in each layer. Pose heatmaps are encoded and inputted into the first style block in StyleGAN2 instead of a constant input.

Unclear: Why does Segmentation and RGB get passed on?

Style Block seems to be just an image at a certain resolution. We could pick a smaller one for efficiency's purpose

3.2. Try-On Optimization

Given the trained model, we can generate a variety of images (along with the corresponding segmentation) within the latent space of the network with the desired 2D pose. Conversely, given an input pair of images, we can "project" the images to the latent space of the generator by running an optimizer to compute the latent codes that minimizes the perceptual distance between the input image and the image from the generator. Linear combinations of these latent codes will produce images that combine various characteristics of the pair of input images. The desired try-on image where the garment from the second image is transferred to the person from the first image lies somewhere within this space of combinations. Let us denote by σ_p and σ_g the style scaling coefficients per layer for person and garment images respectively. Interpolation between the style vectors can be expressed as:

Unclear, but sounds important

Important

Unclear: style scaling coefficients per layer

$$\sigma_t = \sigma_p + Q(\sigma_g - \sigma_p) \quad (1)$$

Used for morphing / merging the two images. A matrix instead of a scalar is used to allow for local editing

where Q is a positive semi-definite diagonal matrix. The elements along the diagonal form a query vector, $q \in [0,1]$. Generating the try-on image can be accomplished by recovering the correct interpolation coefficients q. [6] proposes a greedy algorithm to choose binary query vectors that maximize changes within the region of interest while minimizing changes outside of the region of interest.

We also optimize over the query vectors but instead of greedy search for a set of coefficients using a fixed budget for every layer as in [6], we propose an optimization-based approach that allows for more flexibility in the choice of query vectors. Our loss terms are tuned to the try-on problem of preserving the identity of the person while switching the garment of interest. The loss functions in our optimization guide our method to learn continuous query vectors that enable more localized semantic edits.

Let S_p, S_g, and S_t be the segmentation labels corresponding to I_p, I_g, and I_t. Figure 2 presents the flow of our algorithm. We modify our pose-conditioned StyleGAN2 to take in intermediate latent codes z_1^+ and z_2^+ for both the person and garment images. Style interpolation (Eq. 1) occurs in every style block and the generator outputs the try-on image and segmentation. The outputs are combined to calculate the loss terms which are optimized over the space of interpolation-coefficients q until convergence. Our loss function is defined as follows:

$$L = \alpha_1 L_{localization} + \alpha_2 L_{garment} + \alpha_3 L_{identity} \quad (2)$$

where the αs are weights applied the loss terms and hyperparameters for our method. At each iteration, the q vector values are clipped to $[0, 1]$ using a sigmoid after applying the updates. Each of the loss terms is described below.

Editing-localization Loss The editing-localization loss term encourages the network to only interpolate styles within the region of interest. Similar to [6], we define a term, M, that measures spatial overlap between the semantic regions in the image and the activation tensors, A_{NxCxHxW}, where N is the number of images, C is the number of channels, and H, W are the image dimensions. [6] uses k-means on the activation tensors to assign semantic cluster memberships to the activation tensors. Instead, we use the segmentation outputs from our network to define semantic cluster memberships. The segmentations are converted to binary cluster membership heatmaps, $U \in \{0,1\}^{\text{NxKxHxW}}$, where K is the number of segments. For each layer, the activation tensors are normalized and the heatmaps are downsampled to the correct resolution. M is then computed as:

$$M_{k,c} = \frac{1}{NHW} \sum_{n,h,w} A^2 \odot U \quad (3)$$

M is computed for both the person (I_p) and garment (I_g) images. We then calculate the contribution of each channel within a particular layer for a particular segment of interest, i, at every layer:

$$m_p^i = \max_k (M_p - M_p^i) \quad (4)$$

$$m_g^i = \max_k (M_g - M_g^i) \quad (5)$$

$$m^i = \max(m_p^i, m_g^i) \quad (6)$$

High values in m^i represent the channels that correspond to segments other than i in either image. Since we only want to change the segment of interest, i, we want the interpolation coefficients for all other segments to be low. Therefore the editing-localization loss term is computed as:

$$L_{localization} = \sum m^i \odot q \quad (7)$$

Garment Loss To transfer over the correct shape and texture of the garment(s) of interest, we use VGG embeddings [28, 34] to compute the perceptual distance between the garment areas of the two images. Given the segmentation labels S_g and S_t corresponding to the garment and try-on result images, we compute binary masks for the garment in both images. We apply the mask to the RGB images by element-wise multiplication, followed by blurring with a gaussian filter and downsampling to 256 x 256 before finally computing the garment loss $L_{garment}$ as the perceptual distance between the two masked images.

$$L_{garment} = d(I_g^{\text{Garment Masked}}, I_t^{\text{Garment Masked}}) \quad (8)$$

where $d(\cdot, \cdot)$ measures the perceptual distance by calculating a weighted difference between VGG-16 features.

Identity Loss The identity loss term guides the network to preserve the identity of the person p. We use the hair and face regions of the images as a proxy for the identity of the person. Using the segmentation labels S_p and S_t corresponding to the person image I_p and the try-on image I_t, we compute the identity loss following the same procedure as the garment loss above.

$$L_{identity} = d(I_g^{\text{Identity Masked}}, I_t^{\text{Identity Masked}}) \quad (9)$$

Projection To run our algorithm on real images, we first project the real images into an extended latent space, Z+. We use an optimization to learn a latent vector, z, per layer that results in a final image that best captures the identity and garment details of the original image. The optimization uses a perceptual loss [34] to find the optimal latent vectors.

q's definition. Its [0,1]^d ofc

This can be safely ignored

Seems like loss terms is the only thing for garment

Important! Segmentation labels are needed

How q is learned

Unclear: Only q optimized?

Loss function. Alphas are hyperparameters. See 6.2 for their choice

q vectors are clipped to [0,1]

A_{N x C x H x W} is the activation tensor

Unclear

Unclear what exactly vgg embedding is. Pretrained ones available though, startpage "vgg embedding". They used VGG-16, it seems.

Unclear: What is a binary mask?

Figure 4: Results from our method for **shirt** try-on. Note how try-on works well with different body shapes, and adjusts to the new poses. Each row corresponds to a different try on result. Columns represent person, garment, and result.

Figure 5: Results from our method for **pants** try-on. Note how try-on works well with different body shapes, and adjusts to the new poses. Each row corresponds to a different try on result. Columns represent person, garment, and result.

Figure 6: To experiment with real input images (rather than StyleGAN generated) we have used standard projection algorithm. Here we show typical results of projection on our images. Is it useful to see the effect of projection on the quality of the garment representation, since it directly impacts the final try on result. Improving the projection is independent of our optimization algorithm and is part of future work.

4. Experiments

In this section, we provide implementation details, comparison to related works, ablation study, and results from our method on diverse examples.

Dataset We collected a dataset of people wearing various outfits, and partition it into a training set of 104K images, and a test set of 1600 images. The resolution of all the images is 512x512. The dataset includes people of different body shapes, skin color, height, and weight. Additionally the pepole in our dataset can appear in any pose. We focus in this work on females only, and perform try on for tops and pants.

Implementation details Our conditional StyleGAN2 network was implemented in TensorFlow. We trained it for 25 million iterations, on 8 Tesla v100 GPUs, for 12 days. Once the network was trained, we performed a hyperparameter search for the optimization loss weights. We present all parameters in the supplementary material.

Results Figure 1, Figure 4, and Figure 5 show try-on results produced by our method. Note the diversity of the people wearing the items, how the identity of the person is preserved even though the try on output is synthesized from scratch, and the details on the transferred item (note neck lines, pattern, sleeve length, color). It is also worth noting the garment folds appearing on the new person, since that person might have different body shape, or pose. We present try on of both pants and shirts. Our method is also successful in preserving the skin-tone of the person in the input image though the garment image may contain a person with a different skin-tone. In the case of transferring a shorter sleeve length garment, our method synthesizes the arms appropriately though they were not visible in the input image.

4.1. Comparison to Virtual Try-On Methods

We compare to two state-of-the-art virtual try-on methods with code available: ADGAN [22] and CP-VTON [30]. We use the available pre-trained weights for ADGAN and CP-VTON since they require paired data to train, which we don't have. We compare to both qualitatively and quantitatively.

Image projection on StyleGAN latent space The first step of try-on for our VOGUE method is projecting the garment and person image into the latent space of StyleGAN2. The quality of projection impacts the final try-on images. In Figure 6 we show examples of real images and the corresponding projected images. We use the standard StyleGAN2 projection method extended to the Z+ space as described in 3.2. Note that the projection used is independent

Figure 7: Qualitative comparison with [30] and [22]. Each row represents a different pair of inputs. Note the difference in garment quality, adjustment to difference in body shape, skin color, and pose. VOGUE outperforms the state of the art significantly. **Please refer to supplementary material for more results.**

of our optimization method (see Figure 4 for interpolation without projection). Therefore improving projection as future work would continue to improve our final try-on images.

Qualitative Evaluation Figure 7 compares virtual try-on results produced by our VOGUE method with those produced by the baselines for various shirt and body types. Our method is able to synthesize the correct shape of the shirt and preserve high frequency details. In cases where the target shirt has a shorter sleeve while the person is wearing a longer sleeve, VOGUE is able to accurately synthesize the arms and preserve body shape and skin color. In comparison, ADGAN is unable to synthesize the correct shape of the try-on shirt (e.g. neckline). ADGAN synthesizes the correct color of the shirt and coarse identity information, but is unable to preserve details for both the garment and identity. There are also several artifacts that prevent the try-on image from being photo-realistic. CP-VTON copies the hair and face of the person to the try-on image, but is unable to accurately synthesize the person's body. CP-VTON preserves the color of the garment, however the final try-on is typically blurry.

Quantitative Evaluation We evaluate the results using quantitative measures, Fréchet Inception Distance (FID) [14] and an embedding similarity score [29]. Table 1 shows the FID and embedding similarity results. The experiments were run over 800 images for each algorithm. We can see that our method outperforms others on FID score, which represents photorealism. For calibration, we have also calculated FID scores for a set of real images. The embedding similarity score measures the distance between embeddings of the original garment and the garment in the try-on image. Our method has the highest similarity which reflects our method's ability to preserve the shape and details of the try-on garment.

Model	FID ↓	ES ↑
ADGAN [22]	66.82	0.22
CP-VITON [30]	87.0	0.27
Our Try-on on Real	**32.21**	**0.32**
Real Images	11.83	N/A

Table 1: Quantitative measure of our method and the baselines. We use two metrics to compare the methods and types of images: FID to evaluate photorealism and ES (Embedding similarity) to evaluate quality of try-on or how similar is the result to the input in the garment part.

Figure 8: Failure cases for our method. Our method typically fails when garment detail or pose wasn't represented well in the training dataset.

4.2. Ablation study and failure cases

Figure 8 shows examples of when our VOGUE method fails to correctly synthesize try-on images. Rare poses (not well represented in the data) or garment details cause the appearance of the garment to change when transferred to the target pose. We suspect that the results for those would improve with better representation of diverse garments in the training dataset and subsequently in the latent space. Similarly, as discussed above, projection of real images to latent space has artifacts which affect the try-on result. Once projection is improved, our method will be able to generate true to source results.

Figure 9 shows how our result changes with changes to

Figure 9: Ablation study showing the importance of each loss term in the optimization. The study is done on real images.

Figure 10: Ablation study: we compare greedy search for interpolation coefficients as in [6] to our optimization approach. We observe that details like sleeve length and pattern are preserved much better with the per layer optimization approach. Note that we do not compare directly to [6] since we also modified the StyleGAN architecture to include segmentation and condition on pose.

the loss function. We run this ablation study on real images and demonstrate that each loss term is necessary for a photo-realistic result that preserves garment characteristics and person identity. The localization loss prevents the optimization from editing semantic regions outside of the garment of interest. The identity loss preserves the face and hair. The garment loss transfers the shape, color, and texture of the garment.

Figure 10 demonstrates the difference between greedy search for interpolation parameters as in [6] and per layer optimization (ours). We are not comparing directly to [6] since we also modified the architecture of StyleGAN to include segmentation and condition on pose, however the comparison between greedy search and our optimization is valuable. Our method is able to preserve the shape, color, texture, and details of the region of interest (sleeve length) without affecting the other semantic regions. For example, VOGUE can transfer light colored pants to a person originally wearing dark jeans without lightening the rest of the image. On the other hand, VOGUE can change bordering regions of interest in ways that are consistent with the region of interest being transferred. For example, when transferring a short-sleeved shirt to a person with a longer-sleeved shirt, VOGUE synthesizes skin to show more of the arms in the final try-on image.

5. Discussion

In this paper, we have presented an optimization method for high quality try-on. We use the power of StyleGAN2 and show that it is possible to learn internal interpolation coefficients per layer to create a try-on experience. Our method outperforms the state of the art. We have demonstrated promising results in high resolution on a challenging task of try-on. While promising, our method still fails in cases of extreme poses and underrepresented garments. Similarly, when projection of real images is unsatisfactory it directly affects the interpolation results, since interpolation assumes perfect projection. It is a direction for future research to improve projection of real images onto StyleGAN latent space.

The try-on application is designed to visualize fashion on any person, including different skin tones, body shapes, height, weight, and so on, in the highest quality. However, any deployment of our methods in a real-world setting would need careful attention to responsible design decisions. Such considerations could include labeling any user-facing image that has been recomposed, and matching the distribution of people composed into an outfit to the underlying demographics.

Acknowledgements

We thank Edo Collins, Hao Peng, Jiaming Liu, Daniel Bauman, and Blake Farmer for their support of this work.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? *CVPR*, 2020. 2

[2] Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. *CVPR*, 2020. 2

[3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence 24.4*, pages 509–522, 2002. 2

[4] Arunava Chakraborty et al. S2cgan: Semi-supervised training of conditional gans with fewer labels. *arXiv e-prints*, 2020. 2

[5] Chao-Te Chou et al. Pivtons: Pose invariant virtual try-on shoe with conditional image completion. *Asian Conference on Computer Vision*, 2018. 2

[6] Edo Collins et al. Editing in style: Uncovering the local semantics of gans. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 4, 8

[7] Haoye Dong et al. Towards multi-pose guided virtual try-on network. *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[8] Garoe Dorta et al. The gan that warped: Semantic attribute editing with unpaired data. *CVPR*, 2020. 2

[9] Vincent Dumoulin et al. Feature-wise transformations. *Distill*, 2018. 3

[10] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *CoRR*, 2016. 3

[11] Golnaz Ghiasi et al. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *CoRR*, 2017. 3

[12] Ian Goodfellow et al. Generative adversarial nets. *In Advances in neural information processing systems*, 2014. 3

[13] Xintong Han et al. Viton: An image-based virtual try-on network. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[14] Martin Heusel et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 2017. 7

[15] Jialu Huang, Jing Liao, and Sam Kwong. Unsupervised image-to-image translation via pre-trained stylegan2 network. *arXiv preprint arXiv:2010.05713*, 2020. 2

[16] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR*, 2017. 3

[17] Thibaut Issenhuth, Jérémie Mary, and Jérémie Mary. End-to-end learning of geometric deformations of feature maps for virtual try-on. *arXiv preprint arXiv:1906.01347*, 2019. 2

[18] Surgan Jandial et al. Sievenet: A unified framework for robust image-based virtual try-on. *The IEEE Winter Conference on Applications of Computer Vision*, 2020. 2

[19] Tero Karras et al. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 3

[20] Tero Karras et al. A style-based generator architecture for generative adversarial network. *CVPR*, 2019. 1, 2, 3

[21] Tero Karras et al. Analyzing and improving the image quality of stylegan. *CVPR*, 2020. 1, 2, 3

[22] Yifang Men et al. Controllable person image synthesis with attribute-decomposed gan. *CVPR*, 2020. 2, 6, 7

[23] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[24] Assaf Neuberger et al. Image based virtual try-on network from unpaired data. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[25] Justin NM Pinkney and Doron Adler. Resolution dependant gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020. 2

[26] Amir Raffiee and Michael Sollami. Garmentgan: Photo-realistic adversarial fashion transfer. *arXiv preprint arXiv:2003.01894*, 2020. 2

[27] Amit Raj et al. Swapnet: Garment transfer in single view images. *ECCV*, 2018. 2

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[29] Yang Song et al. Learning unified embedding for apparel recognition. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017. 7

[30] Bochao Wang et al. Toward characteristic-preserving image-based virtual try-on network. *ECCV*, 2018. 2, 6, 7

[31] Ting-Chun Wang et al. High-resolution image synthesis and semantic manipulation with conditional gans. *CVPR*, 2018. 2

[32] Han Yang et al. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. *CVPR*, 2020. 2

[33] Gokhan Yildirim et al. Generating high-resolution fashion model images wearing custom outfits. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 2

[34] Richard Zhang et al. The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*, 2018. 4

6. Supplementary

Below we present additional network and ablation details, as well as more results. **Please refer to the supplementary video** to see a visualization of the results on the same person. It clearly shows how the identity is preserved while shirts and pants switch.

6.1. Implementation Details

The final loss function used for both generated and real image try-on is in Eq. 10.

$$L = \alpha_1 L_{\text{localization}} + \alpha_2 L_{\text{garment}} + \alpha_3 L_{\text{identity}} \tag{10}$$

We performed a hyperparameter search over the loss term weights. For generated images, we used $\alpha_1 = 0.01$, $\alpha_2 = 1$, and $\alpha_3 = 0.2$. For real images, we used $\alpha_1 = 0.01$, $\alpha_2 = 1$, and $\alpha_3 = 1.0$.

The try-on optimization method was run for 2,000 iterations for both real and generated images. The real images were first projected into the StyleGAN2 latent space. The projection optimization was also run for 2,000 iterations per image.

6.2. Additional Results

Figure 11 shows examples of the high-resolution images and the segmentations outputted from our pose-conditioned StyleGAN2. Our network outputs both RGB images and corresponding segmentations with 9 labels (background, tops, bottoms, face, hair, arms, torso/skin, legs, other).

In Figure 12, we demonstrate that the pose-conditioning part of our model is able to synthesize the same style in a variety of poses.

Figure 13 shows additional *shirt try-on* results for generated images. Our method is able to transfer all types of necklines, sleeve shapes, and sleeve lengths for various body types. Additionally, our method is able to preserve identity and synthesize skin correctly when changing from a long to short-sleeved shirt.

In Figures 14 and 15, we show try-on results for real images. These images are first projected into our StyleGAN2 latent space. Projection is a pre-processing step and a direction of future work. It is a current limitation of our method as some of the garment details get lost in projection. Our results show that our method is able to synthesize the correct color and shape of the garments across poses while preserving identity. Improving projection would improve the details and textures, resulting in the quality shown in e.g., Figure 4. The quality of our core method (the network architecture and interpolation optimization) **is best evaluated by looking on generated images**.

Figure 11: We train a pose-conditioned StyleGAN2 network to output segmentations in addition to RGB images. The figure shows our typical generated images and their corresponding segmentation.

Figure 12: Our method can synthesize the *same style shirt* for varied poses and body shapes by fixing the style vector. We present several different styles in multiple poses. In this figure, each row is a fixed style, and each column in a fixed pose and body shape.

Figure 13: Results. We present 6 examples, note the difference in body shape, skin color, types of shirts.

Person Garment **Shirt** try-on

Figure 14: Results from our method for **shirt** try-on on **real** images. See text for explanation of real vs generated images. Note how try-on works well with different body shapes, and adjusts to the new poses. Some details are missing from the garment due to artifacts in projection, however the overall shape is well preserved.

Person Garment **Pants** try-on

Figure 15: Results from our method for **pants** try-on on **real** images. See text for explanation of real vs generated images. Note how try-on works well with different body shapes, and adjusts to the new poses. Some details are missing from the garment due to artifacts in projection, however the overall shape is well preserved.