Wish You Were Here: Context-Aware Human Generation

Oran Gafni
Facebook AI Research
oran@fb.com

Lior Wolf
Facebook AI Research and Tel-Aviv University
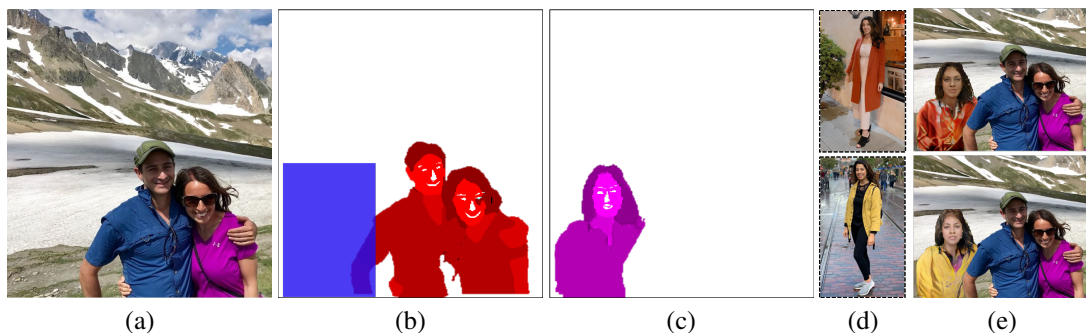wolf@fb.com

| | | | | |
| (a) | (b) | (c) | (d) | (e) |

Figure 1. The "wish you were here" application. Given an image with one or more persons (a), and an optional bounding box indicating where to add a new person (b), the method generates the pose of the new person (c). Then, given the appearance information of the target person (d), the method renders a new image (e) with that person.

Abstract

We present a novel method for inserting objects, specifically humans, into existing images, such that they blend in a photorealistic manner, while respecting the semantic context of the scene. Our method involves three subnetworks: the first generates the semantic map of the new person, given the pose of the other persons in the scene and an optional bounding box specification. The second network renders the pixels of the novel person and its blending mask, based on specifications in the form of multiple appearance components. A third network refines the generated face in order to match those of the target person. Our experiments present convincing high-resolution outputs in this novel and challenging application domain. In addition, the three networks are evaluated individually, demonstrating for example, state of the art results in pose transfer benchmarks.

1. Introduction

The field of image generation has been rapidly progressing in recent years due to the advent of GANs, as well as the introduction of sophisticated architectures and training methods. However, the generation is either done while giving the algorithm an "artistic freedom" to generate attractive images, or while specifying concrete constraints such as an approximate drawing, or desired keypoints.

In other contributions, there is a set of semantic specifications, such as in image generation based on scene graphs, or on free text, yet these are not demonstrated to generate high-fidelity human images. What seems to be missing, is the middle ground of the two: a method that allows some freedom, while requiring adherence to high-level constraints that arise from the image context.

In our work, the generated image has to comply with the soft requirement to have a coherent composition. Specifically, we generate a human figure that fits into the existing scene. Unlike previous work in the domain of human placement, we do not require a driving pose or a semantic map to render a novel person, but rather, we generate a semantic map independently, such that it is suitable to the image context. In addition, we provide rich control over the rendering aspects, enabling additional applications, such as individual component replacement and sketching a photorealistic person. Moreover, we provide for significantly higher resolution results (512×512 vs. a resolution of 176×256 or 64×128 in the leading pose transfer benchmarks), over images with substantial pose variation.

The application domain we focus on is the insertion of a target person into an image that contains other people. This is a challenging application domain, since it is easy to

spot discrepancies between the novel person in the generated image, and the existing ones. In contrast, methods that generate images from scratch enjoy the ability to generate "convenient images".

In addition, images that are entirely synthetic are judged less harshly, since the entire image has the same quality. In our case, the generated pixels are inserted into an existing image and can therefore stand out as being subpar with respect to the quality of the original image parts. Unlike other applications, such as face swapping, our work is far less limited in the class of objects.

Similar to face swapping and other guided image manipulation techniques, the appearance of the output image is controlled by that of an example. However, the appearance in our case is controlled by multiple components: the face, several clothing items, and hair.

Our method employs three networks. The first generates the pose of the novel person in the existing image, based on contextual cues that pertain to the other persons in the image. The second network renders the pixels of the new person, as well as a blending mask. Lastly, the third augments the face of the target person in the generated image in order to ensure artifact-free faces.

In an extensive set of experiments, we demonstrate that the first of our networks can create poses that are indistinguishable from real poses, despite the need to take into account the social interactions in the scene. The first and second networks provide a state of the art solution for the pose transfer task, and the three networks combined are able to provide convincing "wish you were here" results, in which a target person is added to an existing photograph.

The method is trained in an unsupervised manner, in the sense that unlike previous work, such as networks trained on the DeepFashion dataset, it trains on single images, which do not present the same person in different poses. However, the method does employ a set of pretrained networks, which were trained in a fully supervised way, to perform various face and pose related tasks: a human body part parser, a face keypoint detector, and a face-recognition network.

Our main contributions are: (i) the first method, as far as we can ascertain, is the first to generate a human figure in the context of the other persons in the image, (ii) a person generating module that renders a high resolution image and mask, given two types of conditioning, the first being the desired multi-labeled shape in the target image, and the second being various appearance components, (iii) the ability to perform training on a set of unlabeled images "in the wild", without any access to paired source and target images, by utilizing existing modules trained for specific tasks, (iv) unlike recent pose transfer work, which address a simpler task, we work with high resolution images, generating 512×512 images, (v) our results are demonstrated in a domain in which the pose, scale, viewpoint, and severe oc-

clusion vary much more than in the pose transfer work from the literature, and (vi) demonstrating photo realistic results in a challenging and so far unexplored application domain.

Our research can be used to enable natural remote events and social interaction across locations. AR applications can also benefit from the addition of actors in context. Lastly, the exact modeling of relationships in the scene can help recognize manipulated media.

2. Related work

There is considerably more work on the synthesis of novel images, than on augmenting existing views. A prominent line of work generates images of human subjects in different poses [2, 13], which can be conditioned on a specific pose [6, 28, 15]. The second network we employ (out of the three mentioned above) is able to perform this task, and we empirically compare with such methods. Much of the literature presents results on the DeepFashion dataset [17], in which a white background is used. In the application we consider, it is important to be able to smoothly integrate with a complex scene. However, for research purposes only and for comparing with the results of previous work [18, 23, 10, 32, 8], we employ this dataset.

Contributions that include both a human figure and a background scene, include vid2vid [25] and the "everybody dance now" work [6]. These methods learn to map between a driver video and an output video, based on pose or on facial motion. Unlike the analog pose-to-image generation part of our work, in [25, 6] the reference pose is extracted from a real frame, and the methods are not challenged with generated poses. Our method deals with generated poses, which suffer from an additional burden of artifacts. In addition, the motion-transfer work generates an entire image, which includes both the character and the background, resulting in artifacts near the edges of the generated pose [20, 7], and the loss of details from the background. In our work, the generated figure is integrated with the background using a generated alpha-mask.

Novel generation of a target person based on a guiding pose was demonstrated by Esser et al., who presented two methods for mixing the appearance of a figure seen in an image with an arbitrary pose [10, 9]. Their methods result in a low-resolution output with noticeable artifacts, while we work at a higher resolution of 512p. The work of Balakrishanan et al. also provides lower resolution outputs, which are set in a specific background [2]. In our experiment, we compared against the recent pose transfer work [18, 23, 32].

A semantic map based method for human generation was presented by [8]. Contrary to our method, this work was demonstrated solely on the lower resolution, and lower pose variation datasets of DeepFashion and Market-1501 (176×256 and 64×128). Additionally, the target encoding method in [8] relies on an additional semantic map, identical to the

desired target person, requiring the target person to be of the same shape, which precludes other applications, such as component replacement. Moreover, the previous method requires the pose keypoints, which increases the complexity of the algorithm, and limits the application scope, such as the one that we show for drawing a person.

As far as we know, no literature method generates a human pose in the context of other humans in the scene.

3. Method

Given a source image x, the full method objective is to embed an additional person into the image, such that the new person is both realistic, and coherent in context. The system optionally receives a coarse position for the new person, in the form of a bounding box b. This allows for crude control over the new person position and size, yet still leaves most of the positioning for the algorithm.

We employ three phases of generation, in which the inserted person becomes increasingly detailed. The Essence Generation Network (EGN) generates the semantic pose information of the target person p in the new image, capturing the scene essence, in terms of human interaction. The Multi-Conditioning Rendering Network (MCRN) renders a realistic person, given the semantic pose map p, and a segmented target person, which is given as a multi-channel tensor t. The Face Refinement Network (FRN) is used to refine the high-level features of the generated face f, which requires special attention, due to the emphasis given to faces in human perception.

3.1. Essence generation network

The Essence Generation Network (EGN) is trained to capture the human interaction in the image, and generate a coherent way for a new human to join the image. Given a two-channel semantic map s of the input image x with a varying number of persons, and an optional binary third channel b containing a bounding box, the network generates the two-channel semantic map of a new person p, which is compatible with the context of the existing persons, as seen in Fig. 2, 5.

More precisely: both s and p contain one channel for the person's semantic map, and one face channel, derived from facial keypoints. s pertains to the one or more persons in the input image, while p refers to the novel person. The semantic map, i.e., the first channel of s and p, is reduced to eight label groups, encoded as the values $0, 36, 72, .., 252$. These represent the background (0), hair, face, torso and upper limbs, upper-body wear, lower-body wear, lower limbs, and finally shoes. The choice of this reduced number of groups is used to simplify semantic generation, while still supporting detailed image generation.

The face channel of s and p is extracted by considering the convex hulls over the detected facial keypoints, obtained

by the method of [5]. The third channel b is optional, and contains a bounding box, indicating the approximate size and position of the new person in p. During training, the bounding box is taken as the minimal and maximal positions of the labels in the x and y axes. Both the face and bounding-box channels are binary and have values that are either 0 or 255.

We train two EGN models (EGN and EGN') in parallel to perform the following mapping:

$$p = EGN(s, f, b) \quad \text{or} \quad p = EGN'(s, f) \qquad (1)$$

where EGN obtains one additional input channel in comparison to EGN'. For brevity, we address EGN below. The input tensors are resized to the spatial dimensions of 368×368 pixels. The subsequent networks employ higher resolutions, generating high resolution images. The EGN encoder-decoder architecture is based on the one of pix2pixHD [26] with two major modifications. First, the VGG feature-matching loss is disabled, as there is an uncertainty of the generated person. In other words, given a source image, there is a large number of conceivable options for a new person to be generated, in the context of the other persons in the scene. These relations are captured by the discriminator loss as well as the discriminator feature-matching loss, as both losses receive both (s, f, b) and p. The second modification is the addition of a derivative regularization loss $\mathcal{L}_{\nabla}^{p} = \|p_x\|_1 + \|p_y\|_1$, which is applied over the first channel of p. This loss minimizes the high-frequency patterns in the generated semantic map image.

3.2. Multi-conditioning rendering network

The MCRN mapping is trained to render and blend a realistic person into the input image x, creating a high-resolution (512×512) image o. It is given a conditioning signal in the form of a semantic pose map p, and an input specifying the parts of a segmented person t, see Fig. 3(a). The conditioning signal p, which is generated by the EGN at inference time, is introduced to the decoder part of MCRN through SPADE blocks [19]. This conditioning signal acts as the structural foundation for the rendered person image z, and the corresponding mask m.

The segmented person t is incorporated through the MCRN encoder, which embeds the target person appearance attributes into a latent space. t allows for both substantial control over the rendered person z (e.g. replacing the person's hair or clothing, as seen in Fig. 7, and supplementary Fig. 1,2,3). The segmented structure of t has the advantage over simply passing the image of the target person, in that it does not allow for a simple duplication of the target person in the output image z. This property is important, as during training we employ the same person in both the target output image, and as the input to MCRN.
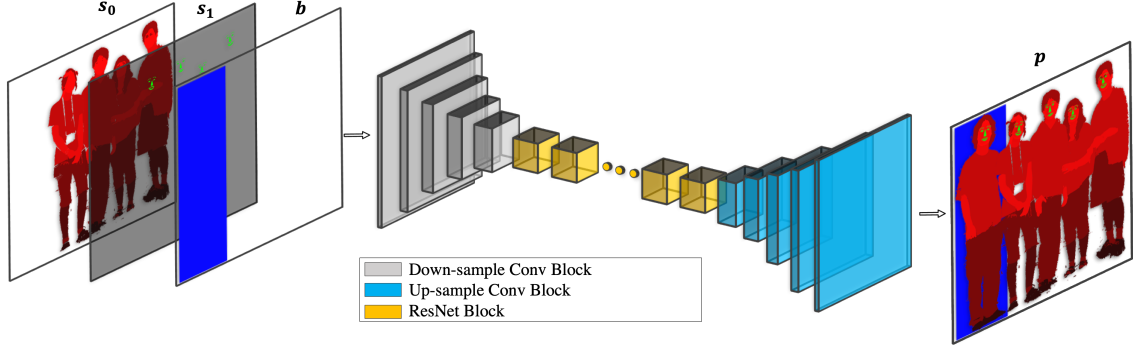
Figure 2. The architecture of the Essence Generation Network. Given a body and face semantic maps s, and an optional bounding-box b, the network generates the semantic map p of a novel person, which is correlated in context to the human interaction in the scene. The generated person is highlighted in blue.

The tensor t is of size $6 \times 3 \times 128 \times 128$, which corresponds to the six semantic segmentation classes (hair, face, upper-body wear, lower-body wear, skin, and shoes), three RGB channels each, and a spatial extent of 128^2 pixels. Each of the six parts is obtained by cropping the body part using a minimal bounding box, and resizing the crop to these spatial dimensions.

To preempt a crude insertion of the generated person z into the image output o and avoid a "pasting" effect, the network generates a learnable mask m in tandem with the rendered image of the person z. The output image is therefore generated as:

$$[z, m] = MCRN(t, p), \quad o = x \odot (1 - m) + z \odot m \quad (2)$$

The mask m is optimized to be similar to the binary version of the pose image p, which is denoted by p^b. For this purpose, the L1 loss is used $\mathcal{L}_1^m = \|m - p^b\|_1$. Additionally, the mask is encouraged to be smooth as captured by the loss

$$\mathcal{L}_\nabla^m = \|m_x\|_1 + \|m_y\|_1. \quad (3)$$

The architecture of the MCRN encoder is composed of five consecutive (Conv2D,InstanceNorm2D [24]) layers, followed by an FC layer with a LeakyReLU activation, resulting in a latent space the size of 256. The latent space is processed through an additional FC layer, reshaped to a size of $4x4x1024$. The decoder has seven upsample layers with interleaving SPADE blocks. It is trained using the loss terms depicted in Fig. 3(b). Namely:

$$\mathcal{L}_{hinge}^G = -\|D_{1,2}(t, p, z^b)\|_1 \quad (4)$$

$$\mathcal{L}_{hinge}^{D_{1,2}} = -\|\min(D_{1,2}(t, p, z^b) - 1, 0)\|_1 - \|\min(-D_{1,2}(t, p, x^b) - 1, 0)\|_1 \quad (5)$$

$$\mathcal{L}_{FM}^{D_{k=1,2}} = \mathbb{E}_{(t,p,x^b,z^b)} \sum_{j=1}^{M} \frac{1}{N_j} \|D_k^{(j)}(t, p, x^b) - D_k^{(j)}(t, p, z^b))\|_1 \quad (6)$$

with M being the number of layers, N_j the number of elements in each layer, $D_k^{(j)}$ the activations of discriminator k in layer j, and $z^b, x^b = z \odot p^b, x \odot p^b$.

$$\mathcal{L}_{FM}^{VGG} = \sum_{j=1}^{M} \frac{1}{N_j'} \|VGG^{(j)}(x) - VGG^{(j)}(o))\|_1 \quad (7)$$

with N_j' being the number of elements in the j-th layer, and $VGG^{(j)}$ the VGG classifier activations at the j-th layer.

$$\mathcal{L}_1^z = \|z^b - x^b\|_1, \mathcal{L}_\nabla^z = \|z_x^b - x_x^b\|_1 + \|z_y^b - x_y^b\|_1 \quad (8)$$

3.3. Face refinement network

The third network, FRN, receives as input the face crop of the novel person in o, as well as a conditioning signal that is the face descriptor of the target face, as obtained from the original image of the target person y (before it was transformed to the tensor t). For that purpose, the pretrained VGGFace2 [4] network is used, and the activations of the penultimate layer are concatenated to the FRN latent space.

FRN applies the architecture of [11], which employs the same two conditioning signals, for a completely different goal. While in [11], the top level perceptual features of the generated face f, obtained from the embedding e of the VGGFace2 network, are distanced from those of the face f_y in y, in our case, the perceptual loss encourages the two to be similar by minimizing the distance $\|e(f) - e(f_y)\|_1$.

FRN's output is blended with a second mask m^f as:

$$[f, m^f] = FRN(c(o), VGGFace2(c(y)))$$
$$w = o \odot (1 - m^f) + f \odot (m^f) \quad (9)$$

where c is the operator that crops the face bounding box.

4. Experiments

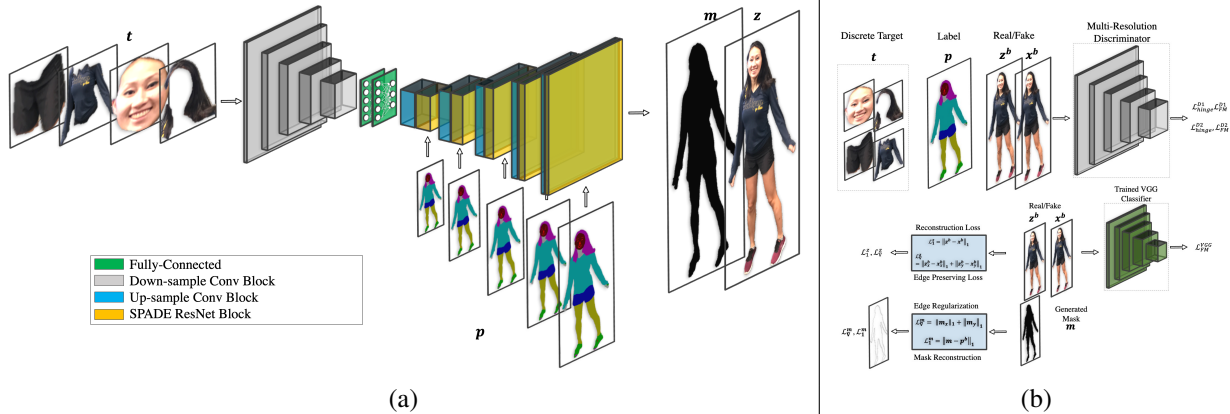Both the EGN and MCRN are trained on the Multi-Human Parsing dataset ([14], [31]). We choose this as our

Figure 3. (a) MCRN's architecture. Given an input target t, and a conditioning semantic map p, a person z and blending mask m are rendered. The mask m is then employed to blend the rendered person z into a final image o. (b) The loss terms used to train MCRN.
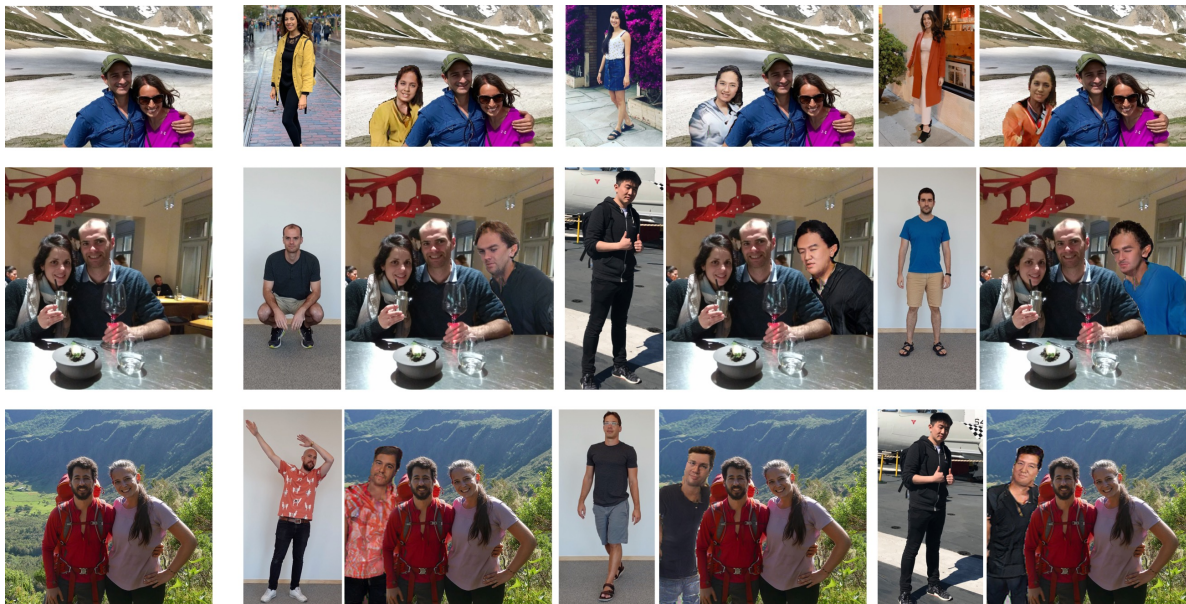

Figure 4. "Wish you were here" samples. Each shows has a source image x, and 3 different pairs of inserted person y and output image w.

primary dataset, due to the high-resolution images and the diverse settings, in terms of pose, scene, ethnicity, and age, which makes it suitable for our task. We randomly select $20,000$ images for training, and 250 images for testing. EGN is trained such that for each sample, all semantic maps in s are maintained, excluding one, which is the generation objective p. In addition, we filter-out images that do not contain at least one detected set of facial keypoints. Overall, we obtain $51,717$ training samples, training for 300 epochs, with a batch size of 64. MCRN is trained on each person separately, resulting in $53,598$ sampled images. The network is trained for 200 epochs, with a batch size of 32.

Our method has a single tuning parameter. This is the strength of the mask edge regularization (Eq. 3) . The scale of the loss term was set during the development process to be multiplied by a factor of 5, after being tested with the

values of $[0, 1, 5, 10]$. This value is verified in the MCRN ablation study in Fig. 8.

Context-aware generation. We provide samples for a variety of target persons y in the full context-aware generation task, in Fig. 4 and supplementary Fig. 1. In these experiments, EGN is given a random bounding-box b, with a size and y-axis location randomly selected to be between 0.9 to 1.1 of an existing person in the image, while the x-axis location is randomly selected by a uniform distribution across the image. The EGN generates a semantic map p, which is then run by the MCRN for various targets y, shown for each column. FRN is then applied to refine the rendered face. As can be observed by the generated results, EGN felicitously captures the scene context, generating a semantic map of a new person that is well-correlated with the human interactions in the scene. MCRN successfully renders a realistic

Figure 5. Unconstrained (no bounding-box) samples of EGN'. For each input (red), the generated pose (purple) is shown.

Figure 6. Drawing a person (DeepFashion). A semantic map is crudely drawn (row 1) utilizing the annotation tool of [31], distinguishing between the hair (orange), face (red), torso/upper-limbs (bright-green 1), T-shirt (yellow), sweat-shirt (bright-green 2), pants (green), lower-limbs (blue). The rendered person generated by the MCRN (row 2) conforms to the conditioning segmentation, despite the deviation from the original dataset. The facial keypoints (not shown here) are taken from a randomly detected image. A video depicting the drawing and generation process is attached in the supplementary.

person, as conditioned by the target y, and blends the novel person well, as demonstrated over diverse targets.

The case with no specified input bounding box is demonstrated in Fig. 5. As can be observed, EGN' selects highly relevant poses by itself.

Individual component replacement. Evaluating the MCRN ability to generalize to other tasks, we utilize it for hair, shirt, and pants replacement, demonstrated over

both the DeepFashion dataset [17] in Fig. 7, supplementary Fig. 2, 3, and the high-resolution Multi-Human Parsing dataset [31] in supplementary Fig. 4. As seen in the latter dataset, MCRN can be successfully applied to unconstrained images, rather than low-variation datasets only, such as DeepFashion, increasing the applicability and robustness of this task. We employ the model of [16, 12] for human parsing.

Person drawing. An additional application of the MCRN is free-form drawing of a person. We intentionally demonstrate this task over a set of extreme, and crudely drawn sketches, depicting the ability to render persons outside of the dataset manifold, yet resulting in coherent results, as seen in Fig. 6, and the supplementary video. The annotation tool presented in [31] is used to sketch the semantic map, and a video depicting the drawing and generation process is attached as supplementary.

Pose transfer evaluation. MCRN can be applied to the pose transfer task. By modifying EGN to accept as input the concatenation of a source semantic map, source pose keypoints (a stick figure, as extracted by the method of [5]), and target pose keypoints, we can generate the target semantic map p, which is then fed into MCRN. A Dense-Pose [21] representation can be used instead of the stick-figure as well.

A qualitative comparison of this pipeline to the methods of [32, 18, 10, 23] is presented in supplementary Fig. 5. The work of [8] presents visually compelling results, similar to ours in this task. We do not present a qualitative comparison to [8] due to code unavailability. However, a quantitative comparison is presented in Tab. 1 (FRN is not applied).

Providing reliable quantitative metrics for generation tasks is well known to be challenging. Widely used methods

Figure 7. Replacing the hair, shirt, and pants (DeepFashion). For each target y (row 1), the hair, shirt and pants (row 2), shirt only (row 3), are replaced for the semantic map s of the upper-left and upper-right person. EGN/FRN are not used. See also Fig. 2,3,4 in the supp.

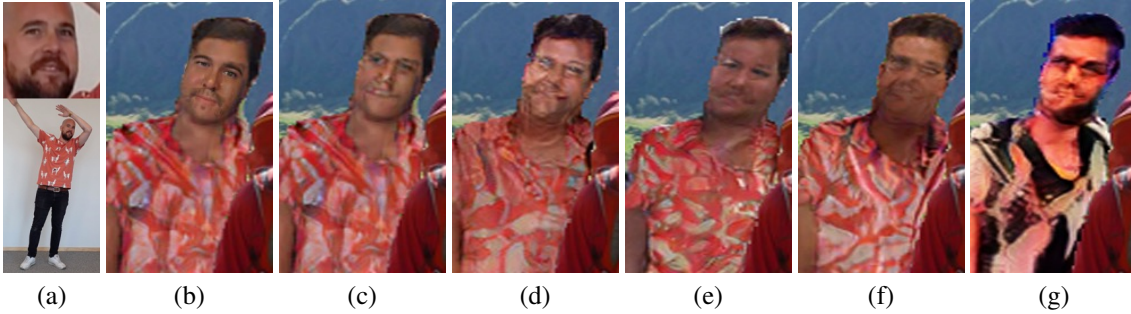   (a)   (b)   (c)   (d)   (e)   (f)   (g)

Figure 8. MCRN ablation study. (a) Target person, (b) our result, (c) no FRN (*distorted face, does not resemble target*), (d) no L_1^z and L_∇^z (*blurry face, distorted skin patterns*), (e) L_∇^m not tuned (*strong edges pixelization*), (f) no mask (*unnatural blending "pasting" effect*), (g) no segmented encoder (*excessive artifacts stemming from target and label spatial difference*).
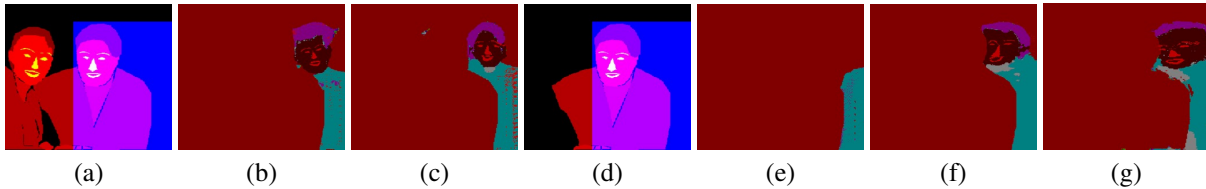

   (a)   (b)   (c)   (d)   (e)   (f)   (g)

Figure 9. EGN ablation study. (a) Semantic map input for (b)-(c), (b) our result, (c) no L_∇^s (*high-frequency patterns, as well as isolated objects generated*), (d) semantic map input for (e)-(g), (e) single person input (*context can be less descriptive*), (f) VGG feature-matching enabled (*shape is matched regardless of deformation artifacts*), (g) generation shape reduced to 256×256 (*labels are perforated, new labels generated on top of existing segmentations*). Columns (b)-(c) and (e)-(g) are presented in high-contrast colors for clarity.

such as Inception Score [22] and SSIM [27] do not capture perceptual notion, or human-structure [3, 32]. Metrics capturing human-structure such as PCK [29], or PCKh [1] have been proposed. However, they rely on a degenerated form of the human form (keypoints).

We therefore develop two new dense-pose based human-structure metrics (DPBS and DPIS), and provide the Python code in the supplementary. Additionally, we evaluate perceptual notions using the LPIPS (Learned Perceptual Image Patch Similarity) metric [30]. DPBS (DensePose Binary Similarity) provides a coarse metric between the de-

tected DensePose [21] representation of the generated and ground-truth images, by computing the Intersection over Union (IoU) of the binary detections. The second novel metric, DPIS (DensePose Index Similarity), provides a finer shape-consistency metric, calculating the IoU of body-part indices, as provided by the DensePose detection. The results are then averaged across the bodyparts.

The quantitative comparison follows the method described by [32] in terms of dataset split into train and test pairs (101,966 pairs are randomly selected for training and 8,570 pairs for testing, with no identity overlap between

Method	↓ LPIPS (SqzNet)	↓ LPIPS (VGG)	↑ DPBS	↑ DPIS	↑ SSIM	↑ IS
Ma [18]	0.416	0.523	0.791	0.392	0.773	3.163
Siarohin [23]	-	-	-	-	0.760	3.362
Esser [10]	-	-	-	-	0.763	**3.440**
Zhu [32]	0.170	0.299	0.840	*0.463*	0.773	3.209
Dong [8]	-	-	-	-	**0.793**	3.314
Ours (DP)	**0.149**	**0.264**	**0.862**	**0.470**	**0.793**	*3.346*
Ours (KP)	*0.156*	*0.271*	*0.852*	0.448	*0.788*	3.189

Table 1. Pose-transfer on the DeepFashion dataset. Shown are the LPIPS [30], DPBS, DPIS, SSIM [27], and IS [22] metrics. Both our DensePose (DP) and keypoint (KP) based methods achieve state-of-the-art results in most metrics. FRN is not applied.

N	Success
3	$39.47\% \pm 0.47$
4	$47.37\% \pm 0.49$
5	$28.07\% \pm 0.43$
6	$47.37\% \pm 0.45$
Average	$42.98\% \pm 0.47$
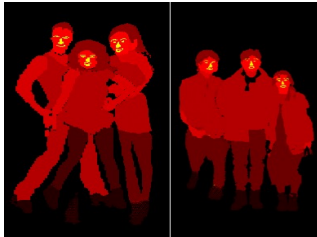
(a)   (b)

Table 2. User study. (a) Success rate in user recognition of the generated person. Shown per N number of persons in an image. (b) Examples of images used. For each image, the user is given unlimited time to identify the generated person.

train and test). Our method achieves the best results in terms of perceptual metrics out of the tested methods (both for our keypoint and DensePose based methods). For human-structural consistency, both our methods achieve top results for the DPBS metric, and highest for the DensePose based model in the DPIS metric. Our methods scores well for the controversial metrics (SSIM, IS) as well.

User Study. A user study is shown in Tab. 2, presented per $N = 3, 4, 5, 6$ number of persons in an image (including the generated person). For each image, the user selects the generated person. The user is aware that all images contain a single generated person, and contrary to user studies commonly used for image generation, no time constraints are given. The low success rate validates EGN's ability to generate novel persons in context. Note that the success rate does not correlate with N as expected, perhaps since the scene becomes more challenging to modify the larger N is.

Ablation study We provide qualitative ablation studies for both EGN and MCRN. As the "wish your were here" application does not have a ground-truth, perceptual comparisons, or shape-consistency quantitative methods do not capture the visual importance of each component. Other methods that do not rely on a ground-truth image (e.g. Inception Score, FID), are unreliable, as for the pose-transfer task, higher IS seems correlated with more substantial arti-

facts, indicating that a higher degree of artifacts results in a stronger perceived diversity by the IS.

The MCRN ablation is give in Fig. 8, showcasing the importance of each component or setting. Details are given in the figure caption.

The EGN ablation is given in Fig. 9. For the generated person, there are numerous generation options that could be considered applicable in terms of context. This produces an involved ablation study, encompassing additional deviations between tested models, that are not a direct result of the different component tested. Observing beyond the minor differences, the expected deviations (as seen throughout the experiments performed to achieve the final network) are detailed in the figure caption.

5. Discussion

Our method is trained on cropped human figures. The generated figure tends to be occluded by other persons in the scene, and does not occlude them. The reason is that during training, the held-out person can be occluded, in which case the foreground person(s) are complete. Alternatively, the held-out person can be complete, in which case, once removed, the occluded person(s) appear to have missing parts. At test time, the persons contain missing areas that are solely due to the existing scene. Therefore, test images appear as images in which the held-out person is occluded.

In a sense, this is exactly what the "wish you were here" application (adding a person to an existing figure) calls for – finding a way to add a person, without disturbing the persons already there. However, having control over the order of the persons in the scene relative to the camera plane, would add another dimension of variability.

A limitation of the current method, is that the generated semantic map p is not conditioned on the target person y or their attributes t. Therefore, for example, the hair of the generated figure is not in the same style as the target person. This limitation is not an inherent limitation, as one can condition EGN on more inputs, but rather a limitation of the way training is done. Since during training we have only one image, providing additional appearance information might impair the network generalization capability. A partial solution may be to condition, for example, on very crude descriptors such as the relative hair length.

6. Conclusions

We demonstrate a convincing ability to add a target person to an existing image. The method employs three networks that are applied sequentially, and progress the image generation process from the semantics to the concrete.

From a general perspective, we demonstrate the ability to modify images, adhering to the semantics of the scene, while preserving the overall image quality.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 7

[2] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8340–8348, 2018. 2

[3] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018. 7

[4] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. *arXiv preprint arXiv:1710.08092*, 2017. 4

[5] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018. 3, 6

[6] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. *arXiv preprint arXiv:1808.07371*, 2018. 2

[7] Patrick Chao, Alexander Li, and Gokul Swamy. Generative models for pose transfer. *arXiv preprint arXiv:1806.09070*, 2018. 2

[8] Haoye Dong, Xiaodan Liang, Ke Gong, Hanjiang Lai, Jia Zhu, and Jian Yin. Soft-gated warping-gan for pose-guided person image synthesis. In *Advances in Neural Information Processing Systems*, pages 474–484, 2018. 2, 6, 8

[9] Patrick Esser, Johannes Haux, Timo Milbich, and Björn Ommer. Towards learning a realistic rendering of human behavior. In *ECCV WORKSHOP*, 2018. 2

[10] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018. 2, 6, 8

[11] Oran Gafni, Lior Wolf, and Yaniv Taigman. Live face de-identification in video. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 4

[12] Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6

[13] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. *arXiv preprint arXiv:1812.01601*, 2018. 2

[14] Jianshu Li, Jian Zhao, Yunchao Wei, Congyan Lang, Yidong Li, Terence Sim, Shuicheng Yan, and Jiashi Feng. Multi-human parsing in the wild. *arXiv preprint arXiv:1705.07206*, 2017. 4

[15] Yining Li, Chen Huang, and Chen Change Loy. Dense intrinsic appearance flow for human pose transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3693–3702, 2019. 2

[16] Xiaodan Liang, Ke Gong, Xiaohui Shen, and Liang Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 6

[17] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 6

[18] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017. 2, 6, 8

[19] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3

[20] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[21] Iasonas Kokkinos Rīza Alp Güler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6, 7

[22] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016. 7, 8

[23] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018. 2, 6, 8

[24] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[25] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[26] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[27] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7, 8

[28] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. *arXiv preprint arXiv:1807.11152*, 2018. 2

[29] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012. 7

[30] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7, 8

[31] Jian Zhao, Jianshu Li, Yu Cheng, Li Zhou, Terence Sim, Shuicheng Yan, and Jiashi Feng. Understanding humans in crowded scenes: Deep nested adversarial learning and a new benchmark for multi-human parsing. *arXiv preprint arXiv:1804.03287*, 2018. 4, 6

[32] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2347–2356, 2019. 2, 6, 7, 8

Supplementary - Wish You Were Here: Context-Aware Human Generation

A. Additional individual component replacement samples
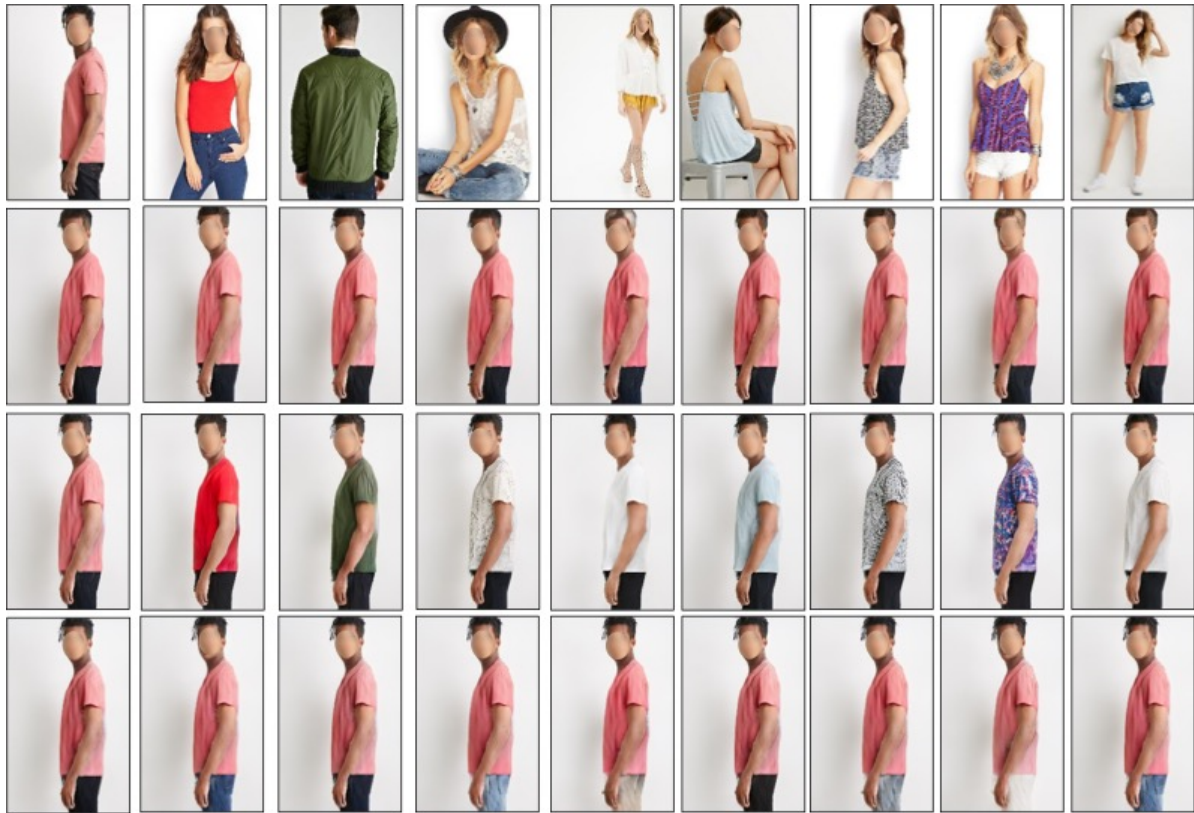

Figure 1: Replacing the hair, shirt, and pants (DeepFashion). For each target y (row 1), the hair (row 2), shirt (row 3), and pants (row 4), are replaced for the semantic map s of the upper-left person. The EGN and FRN are not used.
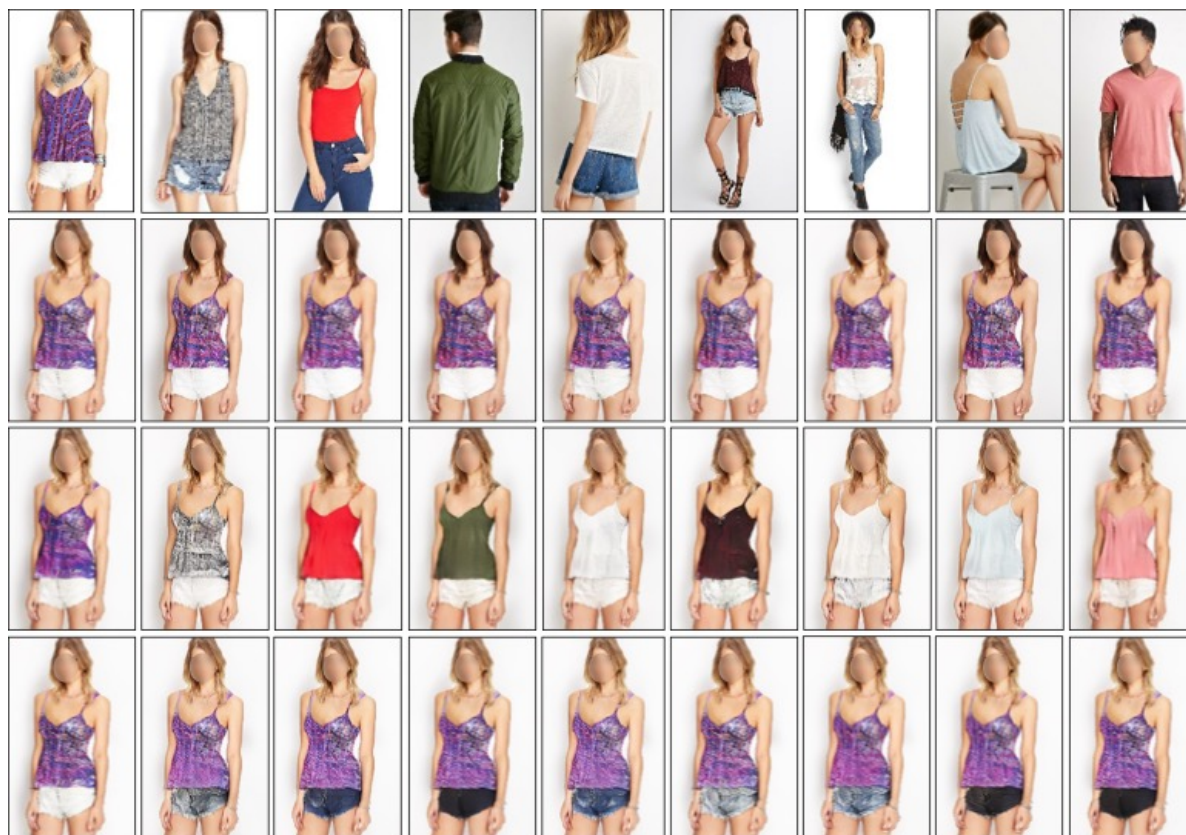
Figure 2: Replacing the hair, shirt, and pants (DeepFashion). For each target y (row 1), the hair (row 2), shirt (row 3), and pants (row 4), are replaced for the semantic map s of the upper-left person. The EGN and FRN are not used.

Figure 3: Replacing the hair, shirt, and pants for high-resolution unconstrained images. For each target y (row 1), the hair (row 2), shirt (row 3), and pants (row 4), are replaced using a chosen semantic map s.

B. Pose-transfer qualitative comparison

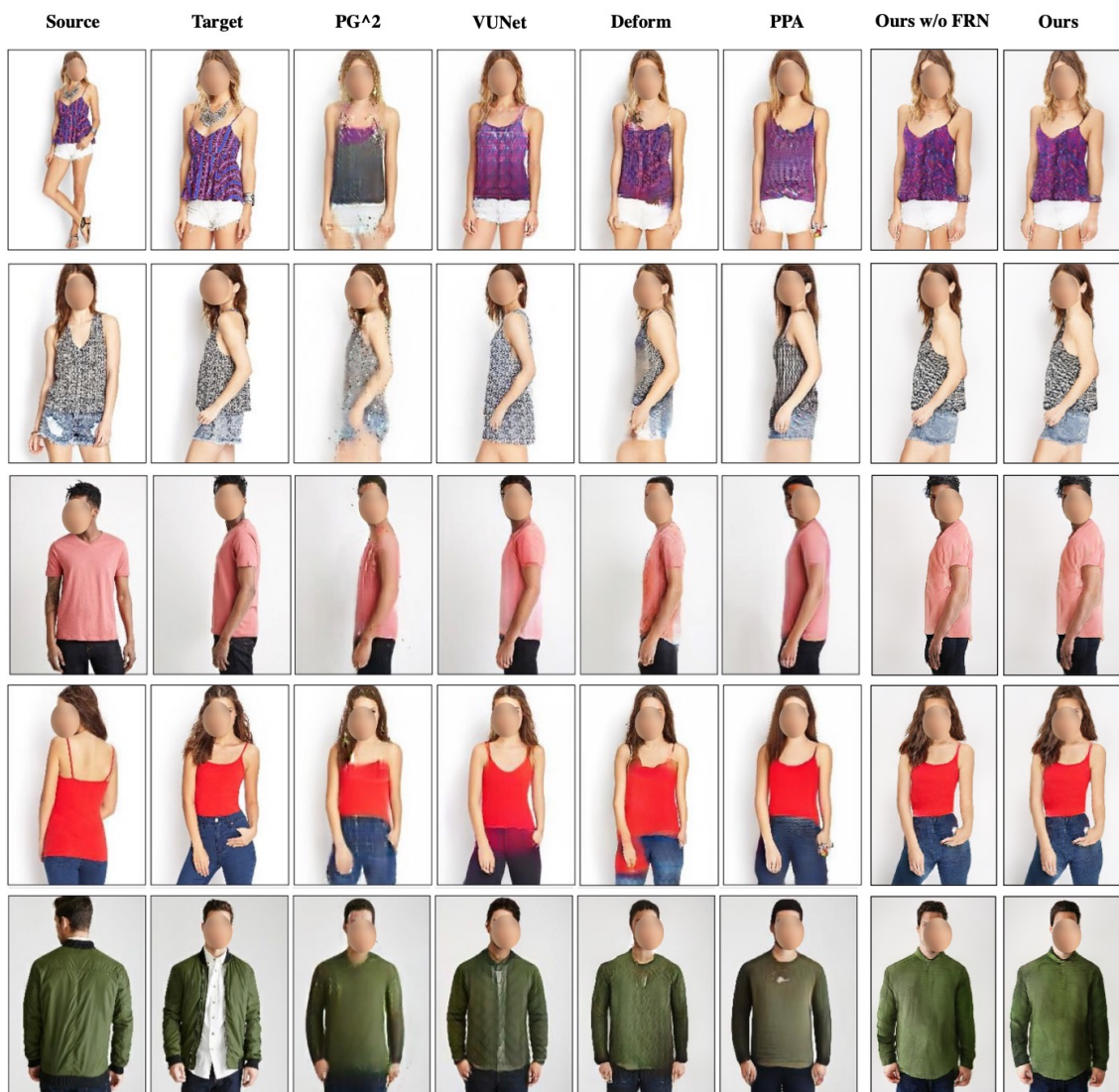| Source	Target	PG^2	VUNet	Deform	PPA	Ours w/o FRN	Ours

Figure 4: Comparison of our method on the pose-transfer task. Even without the Face Refinement Network, our method provides photorealistic rendered targets.
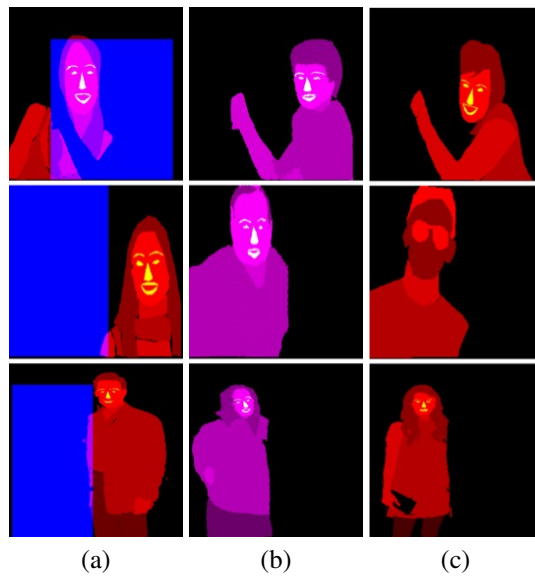
C. EGN training samples

Figure 5: Training the Essence Generation Network. Shown for each row are the (a) input semantic map and bounding box, (b) generated semantic map, (c) ground truth semantic map. The scene essence is captured, while the generated semantic map is not identical to the ground truth.

D. DPBS and DPIS (Python) code

```python
import numpy as np
import os
import cv2

gt_path = 'path/to/ground_truth_densepose'
gen_path = 'path/to/generated_densepose'

read_gen_by_order = True  # Read generated images by order, else by name
n_DP_MAX_IDX = 24  # DensePose generates I with values 0-24

def get_I_iou(img_gt, img_gen, I_idx):
    I_gt = np.zeros_like(img_gt)
    I_gen = np.zeros_like(img_gen)

    I_gt[img_gt == I_idx] = 1  # binarization of the GT image
    I_gen[img_gen == I_idx] = 1  # binarization of the generated image

    I_iou = get_iou(I_gt, I_gen)
    return I_iou

def get_iou(img_gt, img_gen):
    bin_gt = img_gt.copy()
    bin_gen = img_gen.copy()

    bin_gt[bin_gt > 0] = 1  # binarization of the GT image
    bin_gen[bin_gen > 0] = 1  # binarization of the generated image

    bin_union = bin_gt.copy()
    bin_union[bin_gen == 1] = 1  # union over gt and gen (1 where either is present)

    bin_overlap = bin_gt + bin_gen  # overlap of both
    bin_overlap[bin_overlap != 2] = 0  # overlap will be == 2
    bin_overlap[bin_overlap != 0] = 1  # binarization

    union_sum = np.sum(bin_union)
    if union_sum == 0:  # if neither the generated or GT image are present, mask out
        iou = -1
    else:
        iou = np.sum(bin_overlap) / union_sum

    return iou

def get_stats(metric, masked=False):
    if masked:
        return np.ma.mean(metric), np.ma.std(metric), np.ma.median(metric)
    else:
        return np.mean(metric), np.std(metric), np.median(metric)


gt_list = os.listdir(gt_path)  # get ground-truth file names
gt_list.sort()

gen_list = os.listdir(gen_path)  #  get ground-truth files
gen_list.sort()

n_list = len(gt_list)
n_gen = len(gen_list)
if n_list != n_gen:
    print('Error. Ground-truth and generated folders do not contain the same number of images')
    exit(1)
else:
    print('Computing distance metrics over {} images.'.format(n_list))

DPBSs = np.zeros((n_list))  # DensePose Binary Similarity
```

```python
65    DPISs = np.zeros((n_list))   # DensePose Index Similarity
66
67    for img_idx, filename in enumerate(gt_list):
68        img_gt = cv2.imread(os.path.join(gt_path, filename), cv2.IMREAD_UNCHANGED)[:, :, 0]   # DP GT image
69        if read_gen_by_order:
70            img_gen = cv2.imread(os.path.join(gen_path, gen_list[img_idx]), cv2.IMREAD_UNCHANGED)[:, :, 0]
            # DP Generated image read by order
71        else:
72            img_gen = cv2.imread(os.path.join(gen_path, filename), cv2.IMREAD_UNCHANGED)[:, :, 0]   # DP
        Generated image read by name
73
74        max_idx = max(np.amax(img_gt), np.amax(img_gen))   #  the max index is taken as the max between the
        generated and GT
75        if max_idx > n_DP_MAX_IDX:
76            print('Error. The maximum index value was {}. Should not be over 24'.format(max_idx))
77            exit(1)
78
79        DPBSs[img_idx] = get_iou(img_gt, img_gen)   # get DensePose Binary Similarity
80        I_ious = np.zeros((n_DP_MAX_IDX))   # DPIS indices per image
81        I_mask = np.ones_like(I_ious, dtype=bool)   # masking for DPIS indices per image
82        for I_idx in range(1, max_idx + 1):   # iterated over the indices present
83            I_ious[I_idx - 1] = get_I_iou(img_gt, img_gen, I_idx)   # index IoU (per body part)
84        I_mask[I_ious != -1] = 0   # do not mask IoUs found
85        masked_arr = np.ma.array(I_ious, mask=I_mask)   # masked IoUs
86
87        DPISs[img_idx] = np.ma.mean(masked_arr)   # DensePose Index Similarity is calculated over the
        present indices
88
89        if img_idx % 1000 == 0:
90            print('Done with {}/{} images.'.format(img_idx, n_list))
91    DPBSs_mask = np.zeros_like(DPBSs, dtype=bool)   # masking for DPBS
92    DPBSs_mask[DPBSs == -1] = 1
93    masked_DPBSs_arr = np.ma.array(DPBSs, mask=DPBSs_mask)   # masked IoUs
94
95    DPBS_mean, DPBS_SD, DPBS_median = get_stats(masked_DPBSs_arr, masked=True)
96    DPIS_mean, DPIS_SD, DPIS_median = get_stats(DPISs)
97
98    print('----------------------------')
99    print('DPBS')
100   print('Mean: {}, SD: {}, Median: {}'.format(DPBS_mean, DPBS_SD, DPBS_median))
101   print('----------------------------')
102   print('DPIS')
103   print('Mean: {}, SD: {}, Median: {}'.format(DPIS_mean, DPIS_SD, DPIS_median))
104   print('----------------------------')
```