Unselfie: Translating Selfies to Neutral-pose Portraits in the Wild – Supplementary Material

In this supplementary material, we provide additional results, more visual comparisons with prior art and implementation details.

A Results of using off-the-shelf inpainting network for background

In Fig. S1, we provide side-by-side result comparisons of our original pipeline and a slightly modified pipeline using off-the-shelf inpainting network [8] to fill the dis-occluded background holes before feeding the inpainted background (I_{bg} in Figure 6 of the main paper) into our composition stage. In particular, during inference, we use the pre-trained inpainting network to inpaint the holes $H = H_{selfie}$ (marked in black in second column of Fig. S1). Then we apply a matting algorithm [6] to the retrieved neural-pose portrait to create a new hole $H = H_{neutral}$ (third column of Fig. S1) before feeding it together with the coordinate inpainting result (I_{fg} in Figure 6 of the main paper) as input to our composition network. In Fig. S1, the fourth column shows the result of our original pipeline and the fifth column shows the result of our modified pipeline leveraging the inpainting network.

Theoretically, using a separate inpainting network to handle the background separately allows us to harvest the latest advances in image inpainting and focus our composition module exclusively on the synthesis of foreground details and foreground-background transitions. However, there are pros and cons in practice. Pros: the pretrained inpainting network can help reduce the artifacts in the background holes and near the foreground boundaries. For example, in the top row of Fig. S1, the structure of the door on the right side of the image is better synthesized. In the top and middle rows, the artifacts near the arms are also reduced. Cons: occasionally the inpainting network could introduce small artifacts near the foreground boundary. For example, in the bottom row, the inpainting network introduced some grey regions on top of the girl's right shoulder.

B More comparisons

In Fig. S2, we provide more comparisons between our approach and prior approaches, including VUNET [2]. **Ours** is the result from our original pipeline where we manually picked the best one out of results using our top-5 retrieved poses. **Ours w/ inpainted BG** uses off-the-shelf inpainting network as described above. VUNET produces many artifacts in both body and background regions.

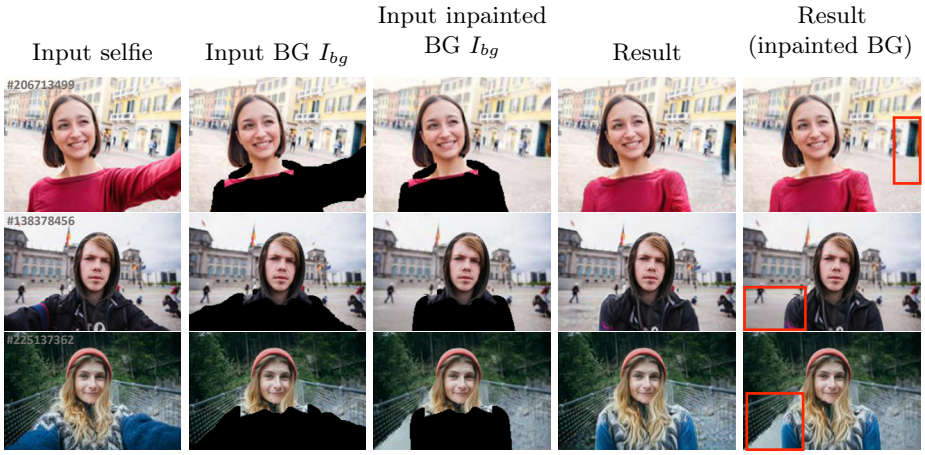

Fig. S1: Results of using inpaited background.

C Multi-modal results

As mentioned in the main paper, our nearest pose search module can generate multiple output variations based on the same input selfie. Fig. S3 provides top5 results for every input selfie. Most of the top5 results have consistent quality with each other. This gives users the freedom to choose the best pose they prefer.

D Implementation details

Image alignment. As mentioned in the main paper, we align the image and pose into the center part of a 256×256 resolution canvas. Likewise, the coordinate-map and texture-map are also in 256×256 resolution. To align the image and pose, we use two shoulder points whose locations are at (63,133) and (92,133) on the 256×256 coordinate-map. After obtaining the coordinates of the two shoulder points from the coordinate-map, we calculate the scale and translation factors for image and pose alignment by aligning the shoulder points to (112,128) and (143,128) on the 256×256 image.

Hyper-parameters and miscellaneous details. For model optimization, we use the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$. $G1$ is trained with a minibatch of size 10 for 70k iterations with initial learning rate of 0.0001. $G2$ is trained with a minibatch of size 2 for 400k iterations with initial learning rate of 0.00002. The loss weights are set to $\lambda_1 = 2$, $\lambda_2 = 10$, $\lambda_3 = \lambda_4 = \lambda_5 = 10$, We use three types of data augmentation during training: 1) left-right image flip; 2) background replacement through foreground mask estimation [6]; 3) random paired selfie selection among top-40 retrieved results. As to the output, we mask out the generated pixels in the invalid region M which denotes the invalid region of the image caused by the alignment step. Therefore, the final output can be formulated

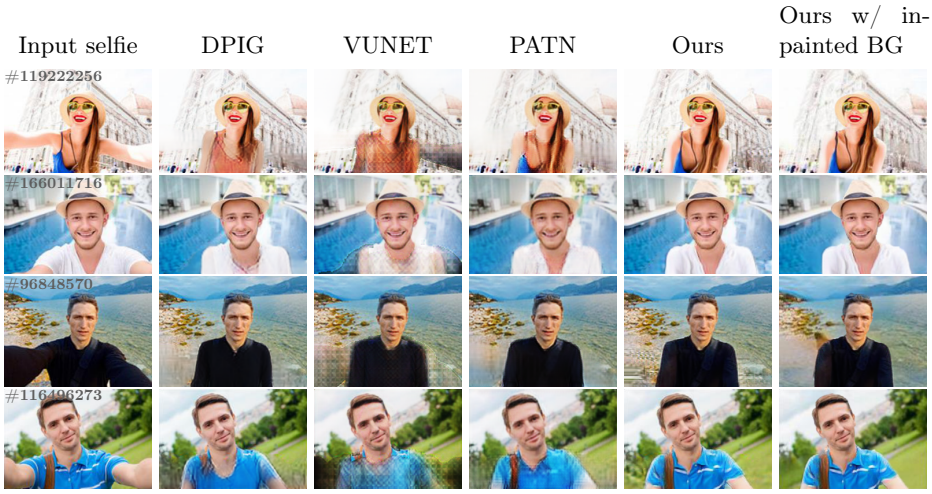

Fig. S2: Comparisons with state-of-the-art methods.

as follows,

$$I_{out} = (I_{G_2} A_{G_2} + I_{bg}(1 - A_{G_2}))(1 - M). \quad (1)$$

The Image2UV (I2UV) mapping is implemented via a lookup table follows [1]

Improvement for DPIG [4] and VUNET [2] As mentioned in the main paper, we made various improvements for DPIG [4] to produce comparable results to ours, because the DPIG model does not converge during training when directly applied to our task. One possible reason is that the background and human appearance in our data contain a lot of variations which are very hard to model in the latent space. For fairer comparison, we improve DPIG in several ways, including adding I_{bg} as input to the decoder, adding perceptual loss [9], using resnet-based PatchGAN discriminator with LSGAN loss [10]. We also improve VUNET by adding I_{bg} as input to U-net encoder and adding L_1 loss to stabilize training. We also tried adding adversarial loss but observe little improvement.

Network architectures. As to our inpainting network architecture, we use the same network as that of [3], except that our input contains 5 channels including C_{src} (2 channels) and T_{src} (3 channels). Our composition network consists of source encoder branch, target encoder branch, and decoder as shown in Tab. S1. The notations $src\_blkN, N = 1, ..., 3$, $tgt\_blkN, N = 1, ..., 3$, $res\_blkN, N = 1, ..., 6$ corresponds to a block with gated convolution layer proposed in [7] followed by group normalization [5] (group number = 32) and Leaky ReLU (slope = 0.01).

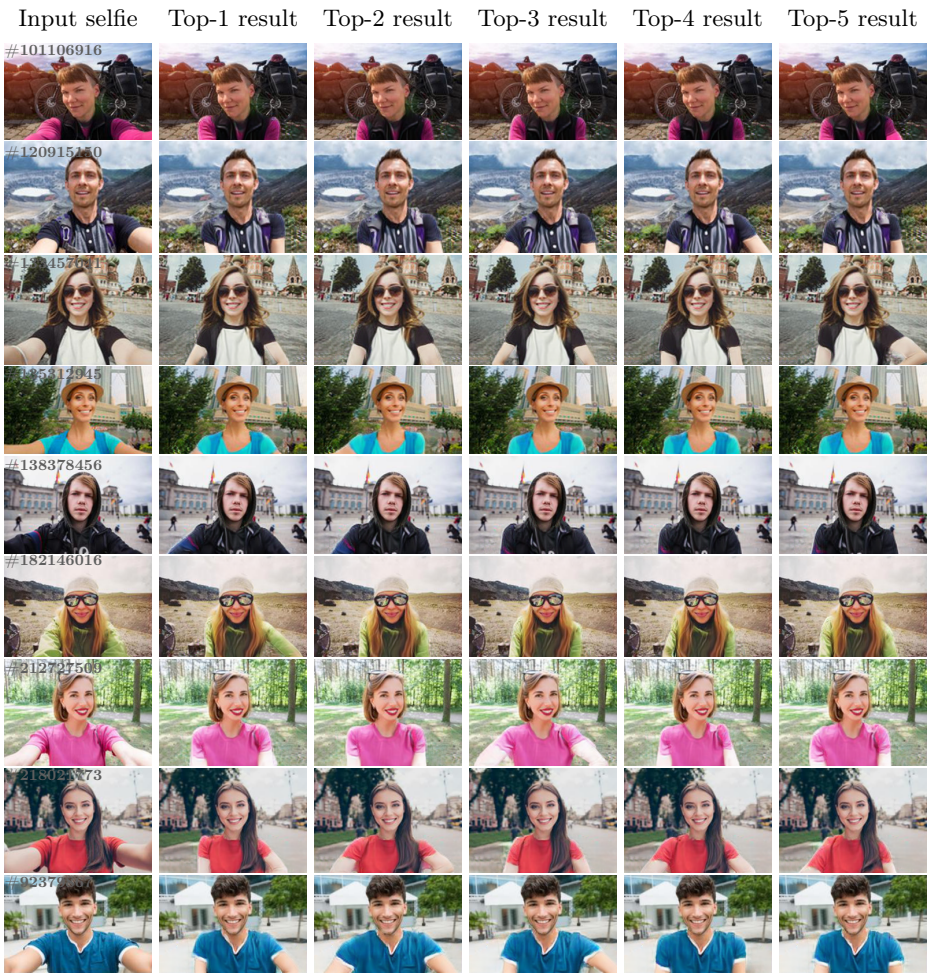

Fig. S3: Top-k results. 1st column: the input selfie image. 2-6th columns: the top-k unselfie results.

Table S1: The composition network architecture.

Layer	Filters/Stride (Dilation)	Input	Input Size	Output Size
Source encoder branch				
src_blk1	5 x 5 / 1 (1)	[P_{src}, I_{fg}]	6 x H x W	256 x H x W
src_blk2	3 x 3 / 1 (1)	src_blk1	256 x H x W	256 x H x W
src_blk3	3 x 3 / 1 (2)	src_blk2	256 x H x W	256 x H x W
Target encoder branch				
tgt_blk1	5 x 5 / 1 (1)	[P_{tgt}, I_{bg}, I_{G1}, M]	10 x H x W	256 x H x W
tgt_blk2	3 x 3 / 2 (1)	tgt_blk1	256 x H x W	256 x $\frac{H}{2}$ x $\frac{W}{2}$
tgt_blk3	3 x 3 / 1 (1)	tgt_blk2	256 x $\frac{H}{2}$ x $\frac{W}{2}$	256 x $\frac{H}{4}$ x $\frac{W}{4}$
res_blk1	3 x 3 / 1 (1)	tgt_blk3	256 x $\frac{H}{4}$ x $\frac{W}{4}$	256 x $\frac{H}{4}$ x $\frac{W}{4}$
res_blk2	3 x 3 / 1 (1)	[res_blk1 + tgt_blk3]	256 x $\frac{H}{4}$ x $\frac{W}{4}$	256 x $\frac{H}{4}$ x $\frac{W}{4}$
res_blk3	3 x 3 / 1 (1)	[res_blk1 + res_blk2]	256 x $\frac{H}{4}$ x $\frac{W}{4}$	256 x $\frac{H}{4}$ x $\frac{W}{4}$
res_blk4	3 x 3 / 1 (1)	[res_blk2 + res_blk3]	256 x $\frac{H}{4}$ x $\frac{W}{4}$	256 x $\frac{H}{4}$ x $\frac{W}{4}$
res_blk5	3 x 3 / 1 (1)	[res_blk3 + res_blk4]	256 x $\frac{H}{4}$ x $\frac{W}{4}$	256 x $\frac{H}{4}$ x $\frac{W}{4}$
res_blk6	3 x 3 / 1 (1)	[res_blk4 + res_blk5]	256 x $\frac{H}{4}$ x $\frac{W}{4}$	256 x $\frac{H}{4}$ x $\frac{W}{4}$
Decoder				
dec_blk1	3 x 3 / 1 (1)	[res_blk5 + res_blk6, warp(src_blk3,E), tgt_blk3]	768 x $\frac{H}{4}$ x $\frac{W}{4}$	256 x $\frac{H}{4}$ x $\frac{W}{4}$
upsample1	—	dec_blk1	256 x $\frac{H}{4}$ x $\frac{W}{4}$	256 x $\frac{H}{2}$ x $\frac{W}{2}$
dec_blk2	3 x 3 / 1 (1)	[upsample1, warp(src_blk2,E), tgt_blk2]	768 x $\frac{H}{2}$ x $\frac{W}{2}$	256 x $\frac{H}{2}$ x $\frac{W}{2}$
upsample2	—	dec_blk2	256 x $\frac{H}{2}$ x $\frac{W}{2}$	256 x H x W
dec_blk3	3 x 3 / 1 (1)	[upsample2, warp(src_blk1,E), tgt_blk1]	768 x H x W	256 x H x W
dec_blk4	3 x 3 / 1 (1)	dec_blk3	256 x H x W	6 x H x W
tanh	—	dec_blk4	6 x H x W	6 x H x W

E Attribution:

Selfie photo owners: #206713499-Paolese, #138378456-iiievgeniy, #225137362-BublikHaus, #119222256-rh2010, #166011716-luengo_ua, #96848570-vitaliymateha, #116496273-travnikovstudio, #101106916-lkoimages, #120915150-wollertz, #133457041-ilovemayorova, #135312945-luengo_ua, #182146016-EVERST, #212727509-Photocatcher, #218021773-deagreez, #92379867-Rido – stock.adobe.com.

References

1. Alldieck, T., Pons-Moll, G., Theobalt, C., Magnor, M.: Tex2shape: Detailed full human body geometry from a single image. In: ICCV (2019)
2. Esser, P., Sutter, E., Ommer, B.: A variational u-net for conditional appearance and shape generation. In: CVPR (2018)
3. Grigorev, A., Sevastopolsky, A., Vakhitov, A., Lempitsky, V.: Coordinate-based texture inpainting for pose-guided image generation. In: CVPR (2019)
4. Ma, L., Sun, Q., Georgoulis, S., Van Gool, L., Schiele, B., Fritz, M.: Disentangled person image generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 99–108 (2018)
5. Wu, Y., He, K.: Group normalization. In: ECCV (2018)
6. Xu, N., Price, B., Cohen, S., Huang, T.: Deep image matting. In: CVPR (2017)
7. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: ICCV (2019)
8. Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: ECCV (2020)
9. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
10. Zhu, Z., Huang, T., Shi, B., Yu, M., Wang, B., Bai, X.: Progressive pose attention transfer for person image generation. In: CVPR (2019)