A Generative Model of People in Clothing
Supplementary Material

Christoph Lassner[1, 2] classner@tue.mpg.de
Gerard Pons-Moll[3,*] gpons@uni-wuerzburg.de
Peter V. Gehler[3,*] peter.gehler@uni-wuerzburg.de

[1]BCCN, Tübingen [2]MPI for Intelligent Systems, Tübingen [3]University of Würzburg

1. Qualitative Results

We complement the qualitative results presented in the main paper by additional ones shown in Fig. 1. The first set in Fig. 1a consists of additional results from ClothNet-full without background completion (*c.f.* main paper, Fig. 9). The model generates plausible people in a diverse set of poses and with a variety of clothes. Larger disconnected components hardly occur, an example can be found in the second row, rightmost picture. Failure cases can observed where the model produces unrealistic body proportions.

In Fig. 1b, we present additional results from ClothNet-body. The first two rows show textured samples from conditioning on the poses presented in the main paper, Fig. 7. Conditioned on a fixed pose and body shape, ClothNet-body generates a variety of clothing types and textures. The third row shows results for a cross-legged pose. Here, the *conditional sketch module* generates both, cross-legged and straight-legged samples with the legs close together. This is due to 3D fitting noise of the body model to the Chictopia10K dataset: cross-legged fits are often erroneous fits to people with straight legs close together and vice versa. The model learns to reproduce this variety observed in the training data. Improved fits would resolve this problem.

In Fig. 1c, we show results for the *portrait* module applied on ground truth sketches from the test set. Most results are plausible, with detailed wrinkles and hair. The model occasionally uses colorful patterns on dresses and tops.

2. Network Architectures

We present the full description of the main CNN architectures in Fig. 2. ClothNet-full is obtained by combining the *latent sketch module* (Fig. 2a) and the *portrait module* (Fig. 2c). It is possible to backpropagate gradients through the entire model, but we trained the parts separately to keep modularity. The *portait* module is combined with the *conditional sketch module* (Fig. 2b) to create ClothNet-body. Encoder and decoder structures are inspired by [1]. Since the portrait module is not variational, we can implement it based on [1] with only slight modifications. In Fig. 2, we show the inputs and outputs for the *latent sketch module* and the *conditional sketch module* with 3 channels. In this configuration, the autoencoders work in the image space of the *plots* of the sketches. The models work with the $256 \times 256 \times 22$ class representation just as well and we include both versions in our code repository. The class representation has the advantage that gradients can be backpropagated through the full ClothNet; this does not work with a model working in the image space because of the plot function is not (trivially) differentiable. Furthermore, we experimented with further hyperparameters and found that the latent vector z is already sufficiently expressive with 32 entries. The published code contains the version with 32 dimensions; to train the models in the paper we used 512 dimensions. The code is available at https://github.com/classner/generating_people.

3. User Study

A standalone and anonymized version of the interface that we used for the user study in Sec. 5.4.2 is part of the supplementary material. A study can be started by opening the html document in any modern browser. Two different studies are available: one to evaluate ClothNet-full and one to evaluate the *portrait* module. Participants were allowed to do both studies in any order. The results of the first ten images are discarded during evaluation to give the participants the opportunity to calibrate for real and fake images. Overall, the feedback for the study was good: users found the task fun; since the task can be completed quickly (150 images with limited display time), users could keep their focus. Participants complained about the low resolution of the images and the short display time. We chose those parameters based on the user study designs presented in [1, 5] for better comparability.

* This work was performed while Gerard Pons-Moll was with the MPI-IS[2]; P. V. Gehler with the BCCN[1] and MPI-IS[2].

(a)

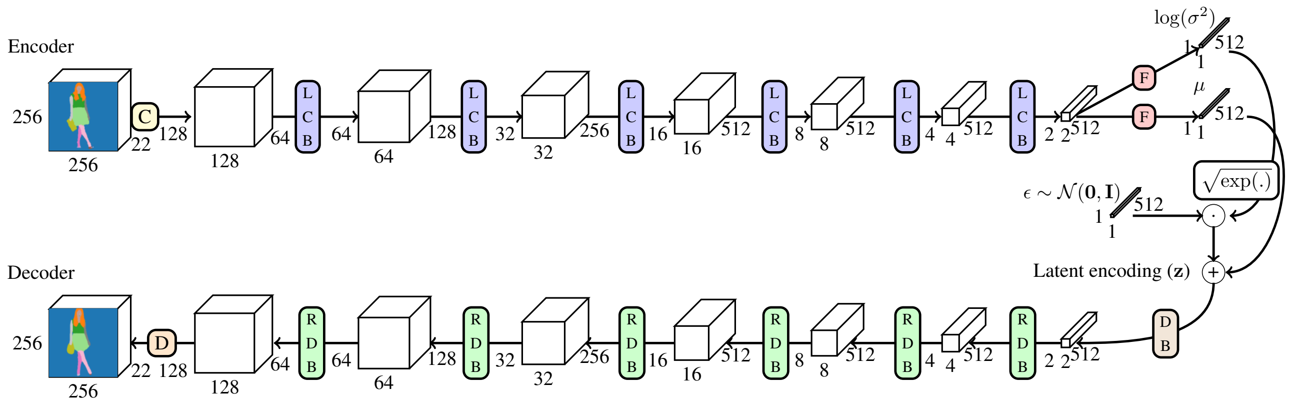
(b)

(c)

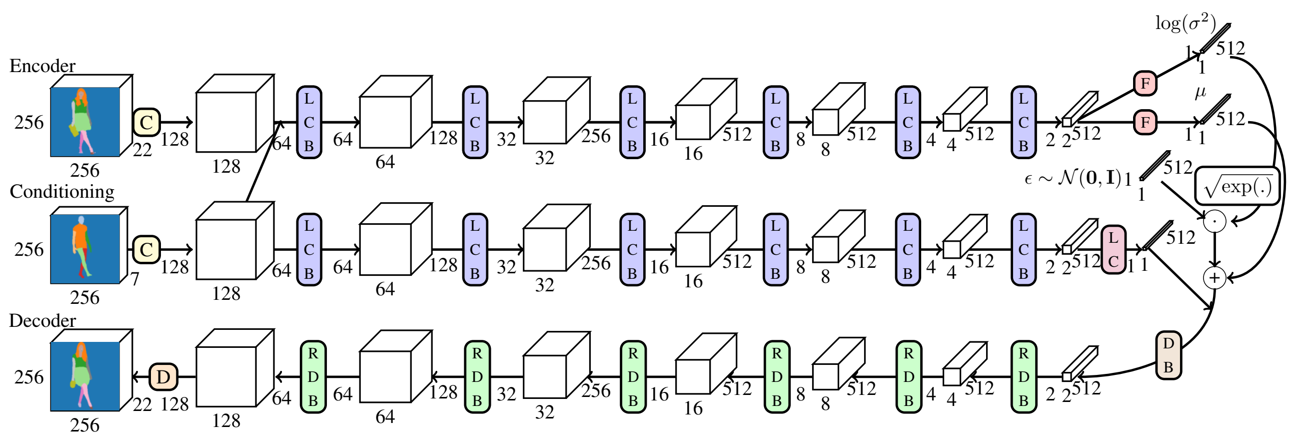
Figure 1: Results from various parts of the proposed model. **(a)** Results from ClothNet-full. **(b)** Results from ClothNet-body. Crossed legs in the third row are sometimes being rendered as crossed or near parallel. This is due to label noise in the training data for the *conditional sketch module* (see Sec. 1 for a full discussion). **(c)** Results from the *portrait* module applied to ground truth sketches from the test set.

4. Combining the VAEs with an adversary

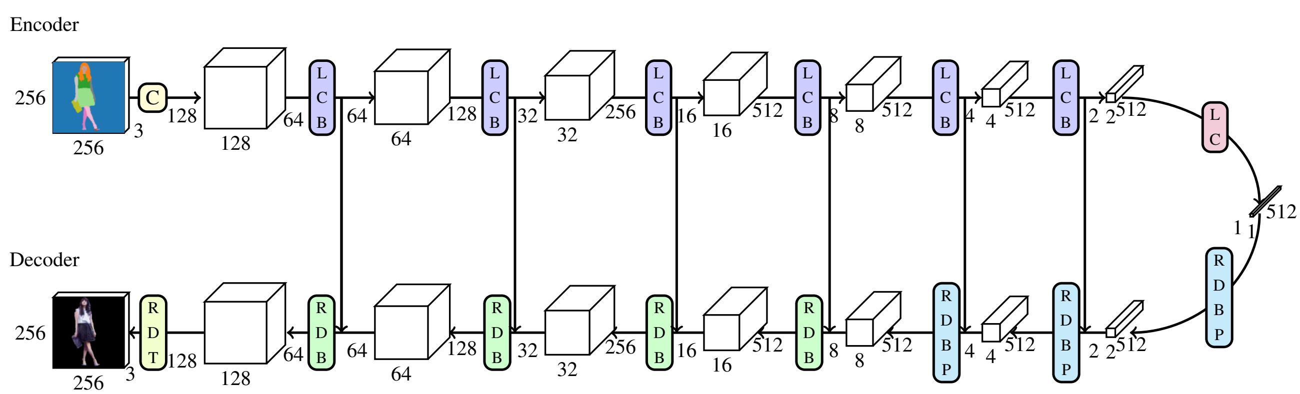
Incorporating adversaries in VAE training is an active topic in the research community [2, 3, 4]. We present results for adding an adversarial loss for the training of the *latent sketch module* in Fig. 3. The loss functions of the variational autoencoder and the adversary must be balanced to prevent the introduction of artificial high frequency structures in the images. With this simple setup, we did not notice striking improvements with the added adversarial loss compared to the balanced variational autoencoder loss.



(a) The *latent sketch module*.



(b) The *conditional sketch module*.



(c) The *portrait module*.

Figure 2: Full configuration of the CNN models presented in the main paper. Data size proportions scale logarithmically. Joining arrows indicate concatenation along the filter axis. The transformation blocks are: C 4×4 convolution, LCB LReLU, 4×4 convolution, batchnorm, F fully connected layer, DB 4×4 deconvolution, batchnorm; RDB ReLU, 4×4 deconvolution, batchnorm; D 4×4 deconvolution; LC LReLU, 4×4 convolution; RDBP ReLU, deconvolution, batchnorm, dropout; RDT ReLU, deconvolution, tanh. The *portrait* module can either be used with color maps as input as depicted here (3 channels) or with probability maps as input (22 channels).

(a)

(b)

Figure 3: Results of the latent sketch module with **(a)** adversary loss weight 1 and **(b)** adversary loss weight 0.01. If the weight is too high, unrealistic artificial high frequency components are added to the sketches to fool the adversary. A lower loss weight helps to counter the effect. With this simple setup, we did not observe striking improvements over VAEs without adversary.

References

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[2] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *ArXiv e-prints 1512.09300*, 2015. 2

[3] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *ArXiv e-prints 1511.05644*, 2015. 2

[4] L. Mescheder, S. Nowozin, and A. Geiger. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. *ArXiv e-prints 1701.04722*, 2017. 2

[5] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 649–666. Springer, 2016. 1