Supplement:
Laplacian Pyramid of Conditional Variational Autoencoders

Garoe Dorta
University of Bath
Anthropics Technology Ltd.
g.dorta.perez@bath.ac.uk

Sara Vicente
Anthropics Technology Ltd.
sara@anthropics.com

Lourdes Agapito
University College London
l.agapito@cs.ucl.ac.uk

Neill D.F. Campbell
University of Bath
n.campbell@bath.ac.uk

Simon Prince
Anthropics Technology Ltd.
simon.prince@anthropics.com

Ivor Simpson
Anthropics Technology Ltd.
ivor@anthropics.com

A NETWORK ARCHITECTURE

Table 1 contains a detailed description of the layers used in the 128×128 model. In order to provide a more intuitive understanding of the architecture, the tensor size after applying each transformation is specified as well. For the 64×64 architecture the same parameters were used, after removing the networks for the 0 level. The parameters for each M network are doubled, where one branch is used to predict the means and the other to predict the variances.

For the network input data, we follow the same procedure as Larsen et al. [1]. The images from the CelebA [2] dataset are centered cropped with a bounding box with top-left and bottom-right corners at [40,15] and [188,163] and downsampled to 64×64 or 128×128. The pixel values are normalized in the [0,1] range, and we augment the data by randomly flipping the images horizontally.

This is an author-prepared pre-print, the definitive version appears in the ACM Digital Library.

Table 1: Network architecture

	Kernel Size	Stride	Output channels	Output Size
Encoder 0				
Convolution Relu	5×5	2	64	$64 \times 64 \times 64$
Convolution Relu	5×5	2	128	$32 \times 32 \times 128$
Convolution Relu	5×5	2	256	$16 \times 16 \times 256$
Fully Connected Relu	-	-	2048	$1 \times 1 \times 2048$
Decoder 0				
Fully Connected Relu	-	-	65536	$16 \times 16 \times 256$
Transposed Convolution Relu	5×5	2	256	$32 \times 32 \times 256$
Transposed Convolution Relu	5×5	2	128	$64 \times 64 \times 128$
Transposed Convolution Relu	5×5	2	32	$128 \times 128 \times 32$
Convolution Sigmoid	5×5	1	3	$128 \times 128 \times 3$
M 0				
Fully Connected Relu	-	-	64	64
Fully Connected Relu	-	-	64	64
Fully Connected Relu	-	-	64	64
Fully Connected Relu	-	-	64	64
Fully Connected	-	-	64	64
Encoder 1				
Convolution Relu	5×5	2	64	$32 \times 32 \times 64$
Convolution Relu	5×5	2	128	$16 \times 16 \times 128$
Convolution Relu	5×5	2	256	$8 \times 8 \times 256$
Fully Connected Relu	-	-	2048	$1 \times 1 \times 2048$
Decoder 1				
Fully Connected Relu	-	-	16384	$8 \times 8 \times 256$
Transposed Convolution Relu	5×5	2	256	$16 \times 16 \times 256$
Transposed Convolution Relu	5×5	2	128	$32 \times 32 \times 128$
Transposed Convolution Relu	5×5	2	32	$64 \times 64 \times 32$
Convolution Sigmoid	5×5	1	3	$64 \times 64 \times 3$
M 1				
Fully Connected Relu	-	-	64	64
Fully Connected Relu	-	-	64	64
Fully Connected Relu	-	-	64	64
Fully Connected	-	-	64	64
Encoder 2				
Convolution Relu	5×5	2	64	$16 \times 16 \times 64$
Convolution Relu	5×5	2	128	$8 \times 8 \times 128$

Table 1: Network architecture

	Kernel Size	Stride	Output channels	Output Size
Convolution Relu	5×5	2	256	$4 \times 4 \times 256$
Fully Connected Relu	-	-	2048	$1 \times 1 \times 2048$
Decoder 2				
Fully Connected Relu	-	-	4096	$4 \times 4 \times 256$
Transposed Convolution Relu	5×5	2	256	$8 \times 8 \times 256$
Transposed Convolution Relu	5×5	2	128	$16 \times 16 \times 128$
Transposed Convolution Relu	5×5	2	32	$32 \times 32 \times 32$
Convolution Sigmoid	5×5	1	3	$32 \times 32 \times 3$
M 2				
Fully Connected Relu	-	-	64	64
Fully Connected Relu	-	-	64	64
Fully Connected	-	-	64	64
Encoder 3				
Convolution Relu	5×5	2	64	$8 \times 8 \times 64$
Convolution Relu	5×5	2	128	$4 \times 4 \times 128$
Convolution Relu	5×5	2	256	$2 \times 2 \times 256$
Fully Connected Relu	-	-	2048	$1 \times 1 \times 2048$
Decoder 3				
Fully Connected Relu	-	-	1024	$2 \times 2 \times 256$
Transposed Convolution Relu	5×5	2	256	$4 \times 4 \times 256$
Transposed Convolution Relu	5×5	2	128	$8 \times 8 \times 128$
Transposed Convolution Relu	5×5	2	32	$16 \times 16 \times 32$
Convolution Sigmoid	5×5	1	3	$16 \times 16 \times 3$
M 3				
Fully Connected Relu	-	-	64	64
Fully Connected	-	-	64	64
Encoder 4				
Fully Connected Relu	-	-	512	512
Fully Connected Relu	-	-	512	512
Fully Connected Relu	-	-	256	256
Fully Connected Relu	-	-	256	256
Decoder 4				
Fully Connected Relu	-	-	256	256
Fully Connected Relu	-	-	256	256
Fully Connected Relu	-	-	512	512
Fully Connected Relu	-	-	512	512
Fully Connected Sigmoid	-	-	192	$8 \times 8 \times 3$

Table 1: Network architecture

	Kernel Size	Stride	Output channels	Output Size

B ANALYSIS OF λ WEIGHTS

The regularizer term in the error function plays an important role in the quality of the reconstructions and the samples produced by the model. A comparison of the effects of using different λ_0 values in the 64×64 model is shown in Figure 1. The "VAE Prior" row denotes an architecture where each level in the Laplacian pyramid is trained with the original VAE prior, instead of our latent space cost. The samples generated without M_k functions highlight the need for a more complex distribution in the latent space, as they are blurry and oversmoothed. The results using M_k show that as λ_0 increases the number of artifacts present in the samples decreases, but the quality of the reconstructions is lowered as well.

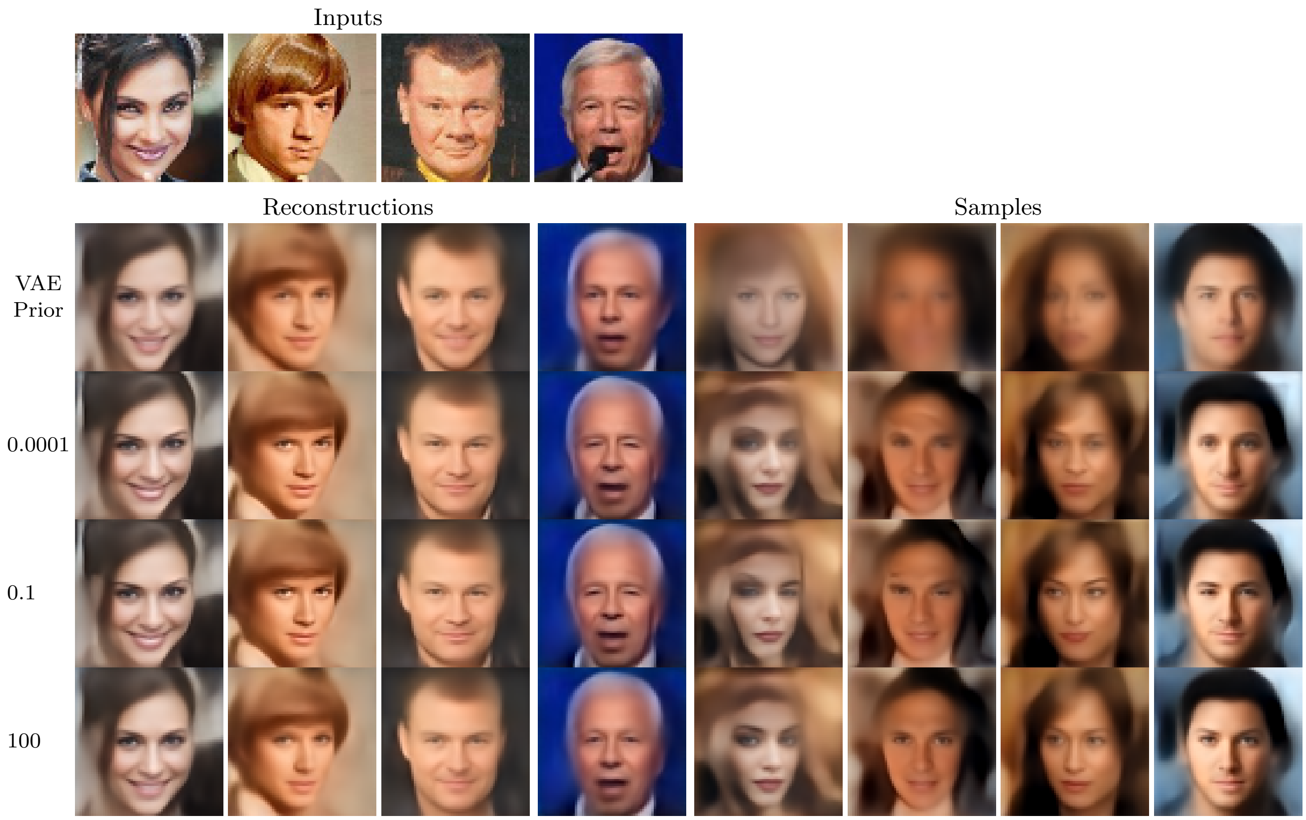

Figure 1: The effect of using different values for the regularizer weight λ, and using the zero mean, unit variance Gaussian prior instead of the M_k functions. Inputs (top-most row), first row correspond to reconstructions and samples from our model without M_k functions, the following rows correspond to $\lambda_0 = [0.0001, 0.1, 100]$, where λ_k for all $k \neq 0$ are fixed to the same value. Without M_k the images are significantly blurry, specially the samples, and when using M_k as λ_0 increases so does the sample quality, yet the reconstructions deteriorate.

C ADDITIONAL RESULTS

A large number of samples and reconstructions using the 128×128 LapCVAE model are shown in Figures 2 and 3.

The total number of samples used to generate Figures 1 and 9 in the paper are shown in Figures 4, 5, 6 and 7. For each editing example shown in the paper, the 32 samples shown here were generated, and the best four were chosen.

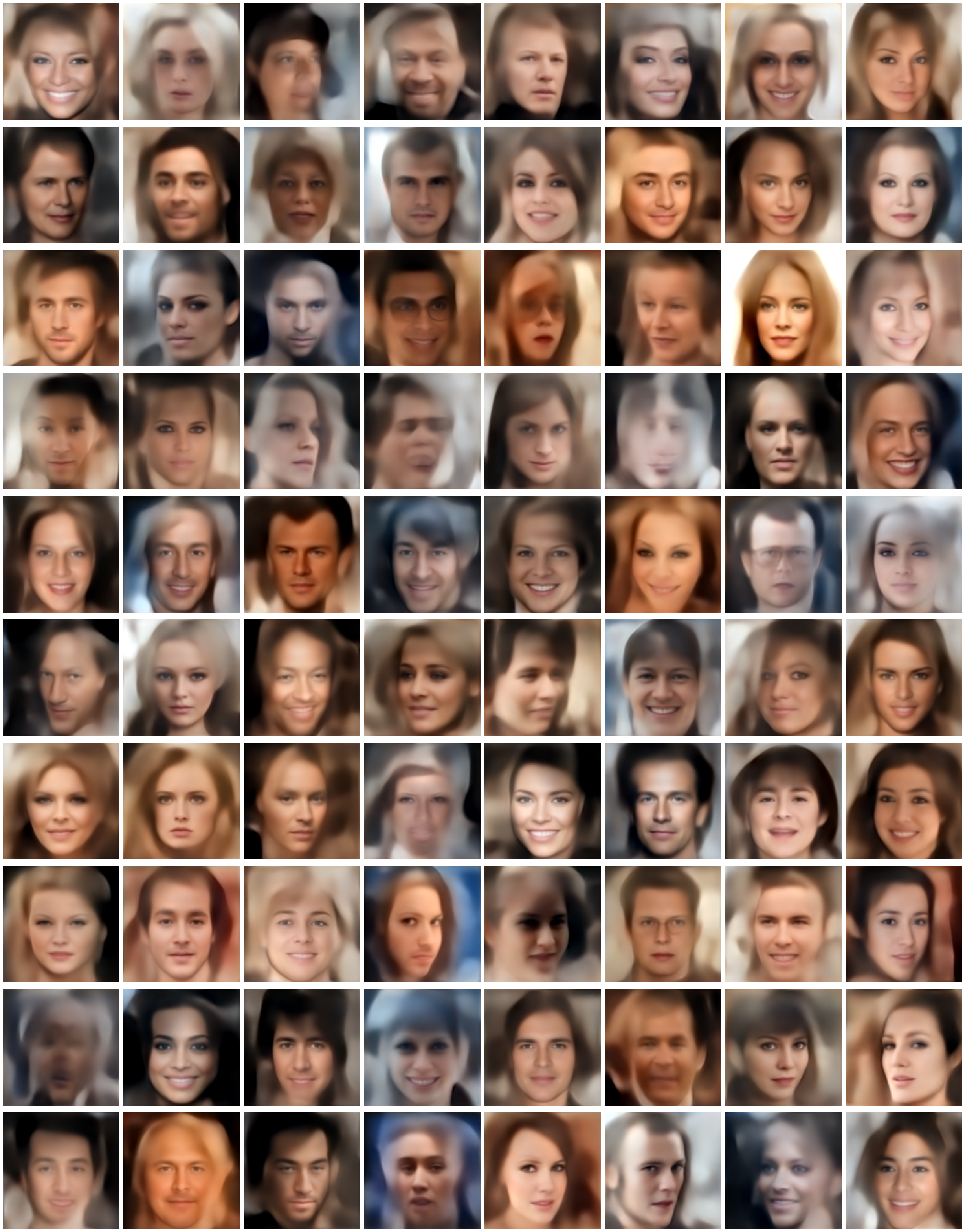

Figure 2: Samples produced by LapCVAE 128×128 after fine-tuning the M_k networks.

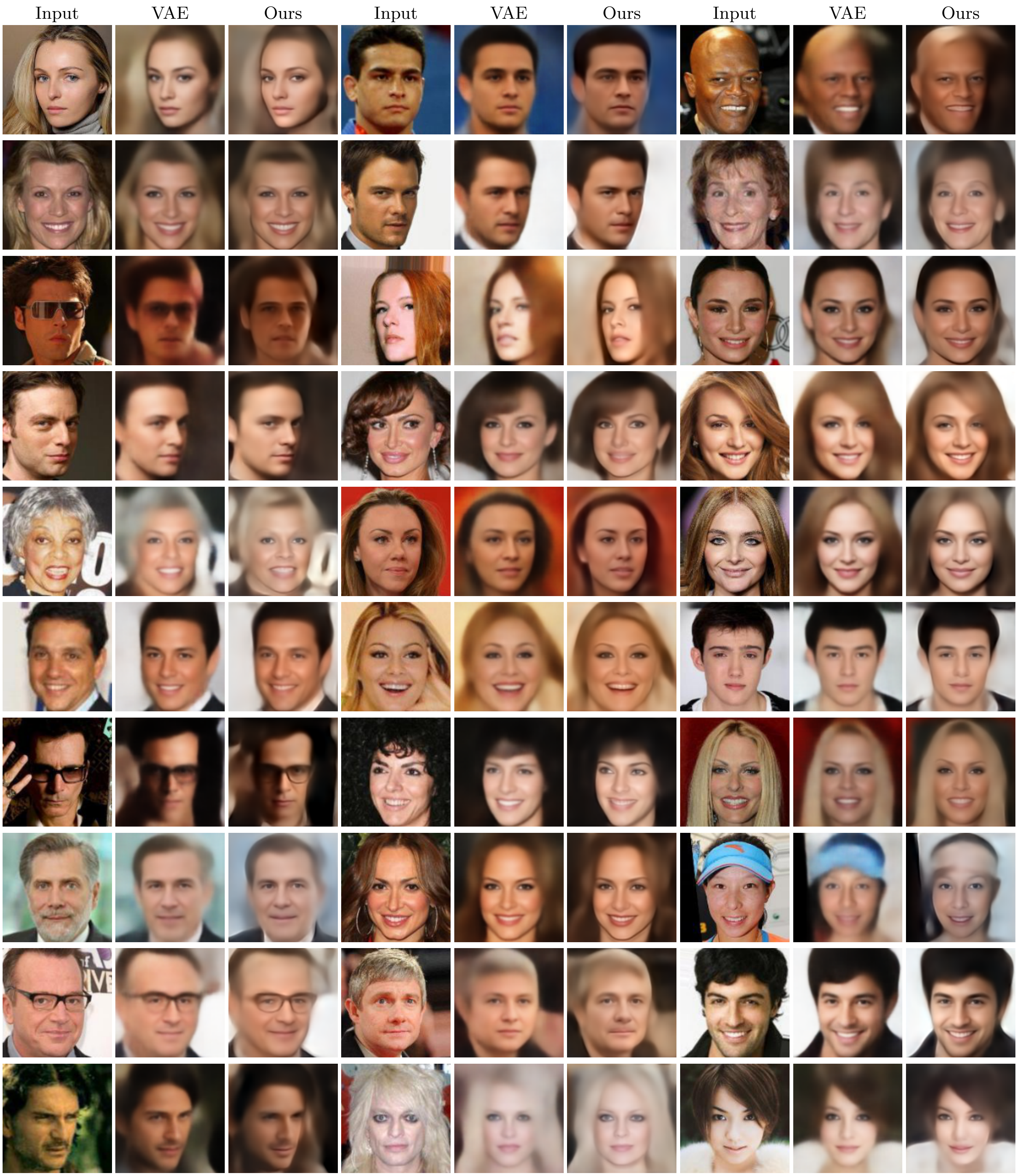

Figure 3: Reconstructions on the 128×128 test data for VAE and our model.

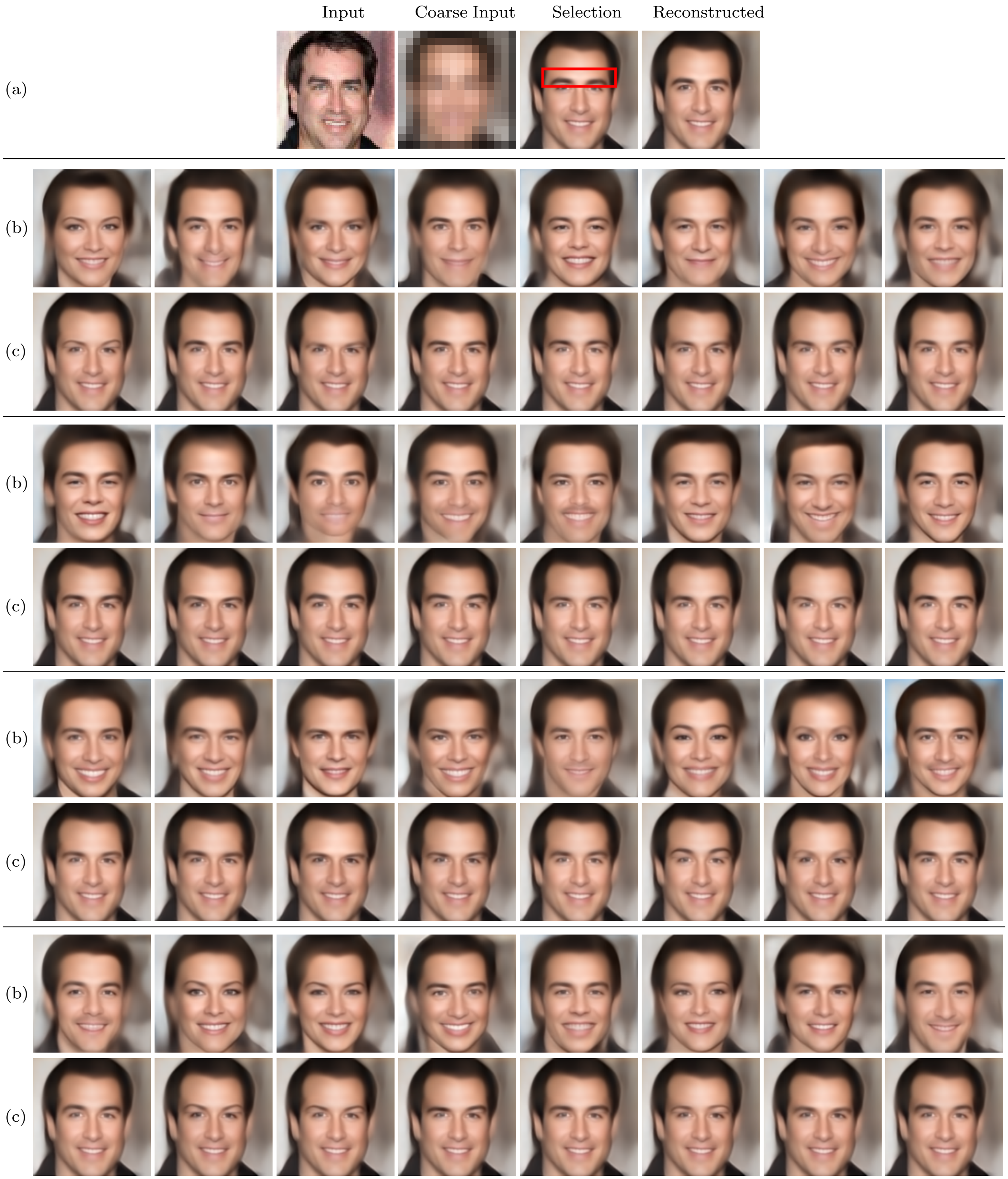

Figure 4: Image editing example 2. (a) An input image is decomposed using a Laplacian pyramid. The user selects an area to be edited. (b) Several samples are generated conditioned on this. (c) The selected regions are composed with the original reconstructed image.

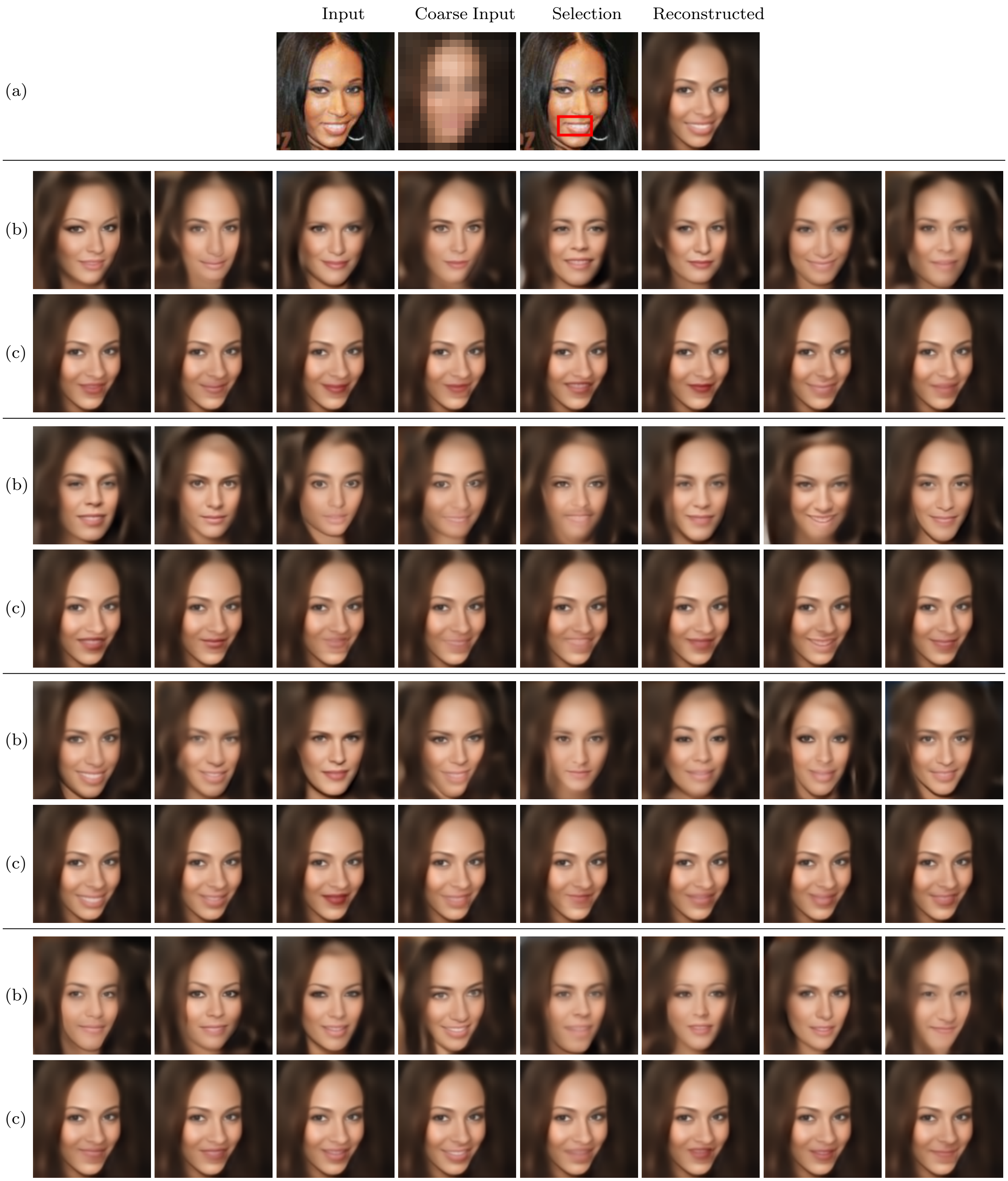

Figure 5: Image editing example 2. (a) An input image is decomposed using a Laplacian pyramid. The user selects an area to be edited. (b) Several samples are generated conditioned on this. (c) The selected regions are composed with the original reconstructed image.

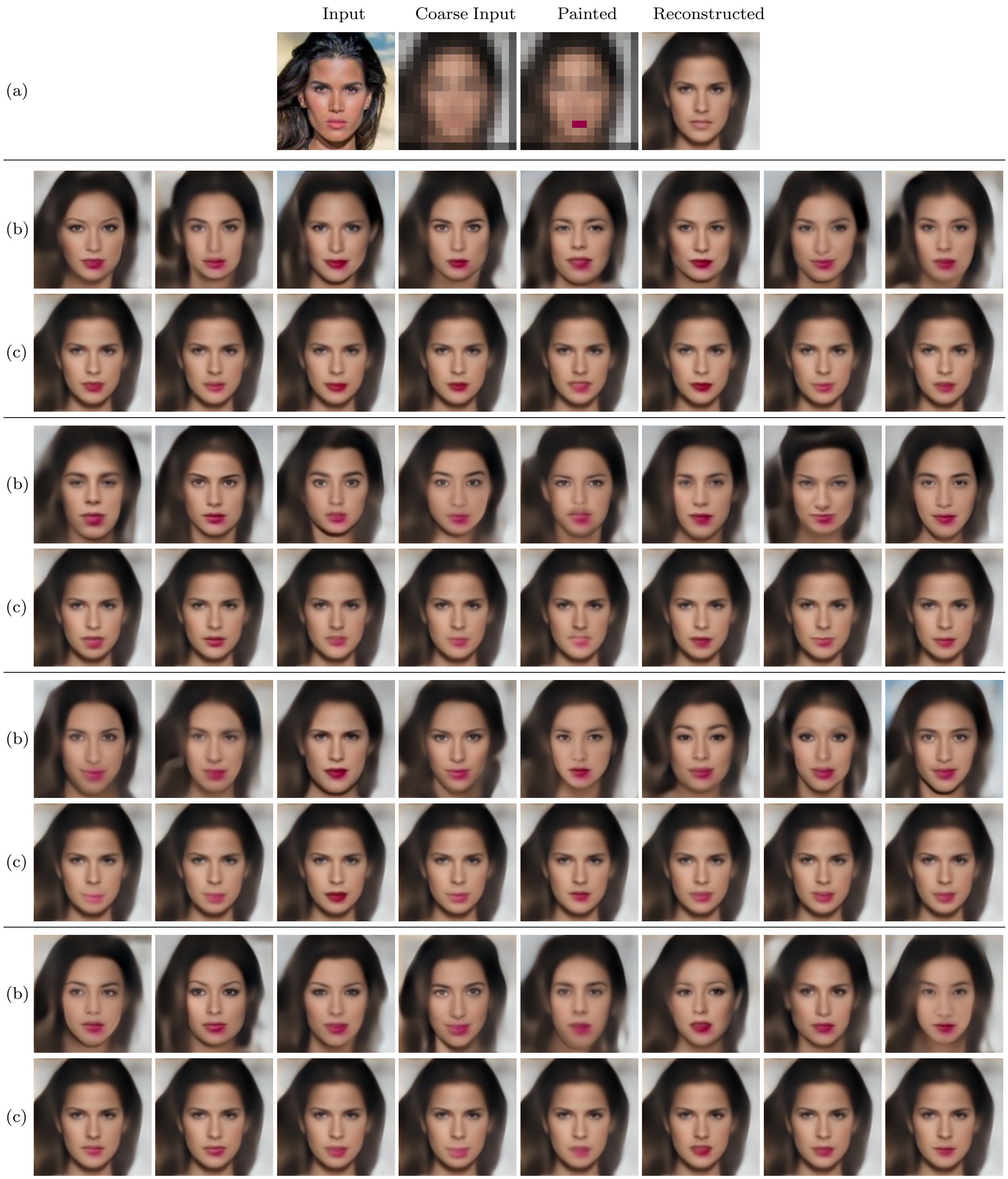

Figure 6: Image editing example 2. (a) An input image is decomposed using a Laplacian pyramid. The user paints into the coarse image. (b) Several samples are generated conditioned on this. (c) The selected regions are composed with the original reconstructed image.

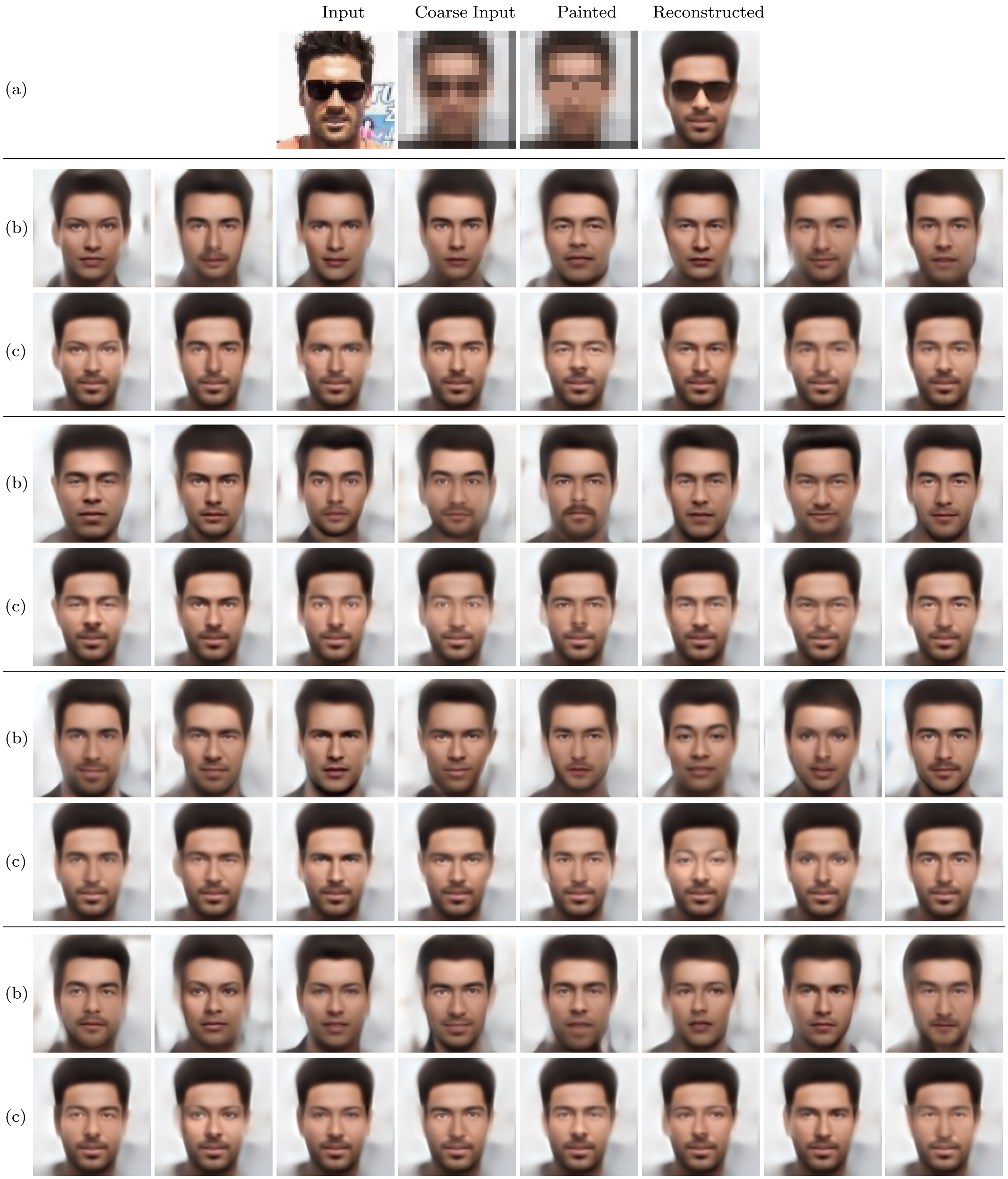

Figure 7: Image editing example 2. (a) An input image is decomposed using a Laplacian pyramid. The user paints into the coarse image. (b) Several samples are generated conditioned on this. (c) The selected regions are composed with the original reconstructed image.

REFERENCES

[1] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. 2016. Autoencoding beyond pixels using a learned similarity metric. In *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48. JMLR, 1558–1566.

[2] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.