6 Supplementary

6.1 High-resolution image generation

While, for simplicity and efficiency, all the experiments are performed on 256×256 pixel images, it is straightforward to increase the output resolution by adding additional layers to the generator and the image discriminators. Figure 9 shows the results of ConfigNet trained at a resolution of 512×512 pixels.

6.2 Comparison to FaceID-GAN

We perform a qualitative comparison to FaceID-GAN [34] by fine-tuning ConfigNet on a number of images from the FFHQ validation set and generating a variety of poses and expressions to match a figure shown in the FaceID-GAN paper. The results of this comparison are shown in Figure 10. While both methods produce consistent poses and expression and preserve the identity well, ConfigNet outputs images with more consistent illumination and background. This is best exemplified in the third row of Figure 10b where the direction and intensity of illumination stay constant despite change of head pose and expression.

6.3 Comparison to CycleGAN

We train CycleGAN [45] to convert images from the synthetic domain (SynthFace [3]) to the real domain (FFHQ [17]). We then generate synthetic image pairs I_+ and I_-, where all the parameters except the modified attribute v are randomly generated and identical between the two images. These images are then passed through CycleGAN and an attribute predictor trained on CelebA [23] to generate the controllability metrics: $C_{pred}(I_+), C_{pred}(I_-)$ and MD (Section 4.1). While CycleGAN and ConfigNet are not fully comparable, as the former does not allow for modifying existing real face images, this simple procedure allows us to compare the two methods in terms of the level of control over images in the real domain.

Table 3 shows the controllability metric results for ConfigNet, CycleGAN and the synthetic data generated with the procedure mentioned above. CycleGAN obtains a slightly better dynamic range $C_{pred}(I_+) - C_{pred}(I_-)$ than ConfigNet, at a cost of a larger mean absolute difference of other attributes. While the difference in MD may look small in absolute terms, the relative increase compared to ConfigNet is 45%. This leads us to the conclusion that while CycleGAN can preserve the very large difference between the values of the modified attribute v in I_+ and I_-, it does not preserve other, more subtle, attributes that should have remained constant. Figure 11 strenghtens this conclusion by showing the effects of hair colour change performed with CycleGAN and ConfigNet, where the former clearly changes other face attributes as well.

The controllability metric results produced by unmodified synthetic data show lowest MD, which is expected, but also lowest dynamic range. We believe

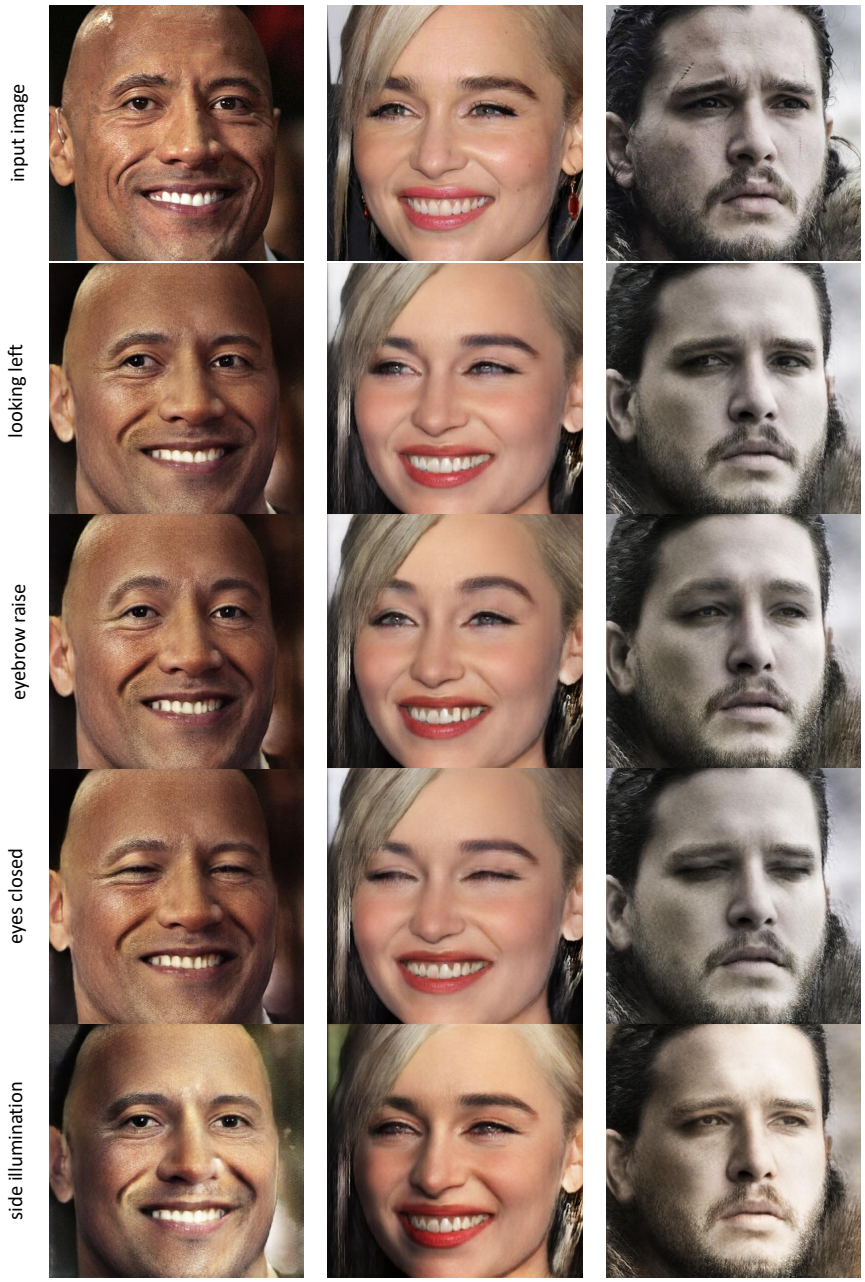

Fig. 9: Results of ConfigNet trained at a resolution of 512×512 pixels. First row shows input image, while the remaining rows show outputs of ConfigNet fine-tuned on the input image. For expression modification we use the fine-grained control procedure (Section 3.6) that allows to modify a subset of the expressions.

(a)

(b)

Fig. 10: Qualitative comparison of a) FaceID-GAN [34] and b) ConfigNet. First column shows the input image, while the remaining columns show varying pose and expression. The identities in a) come from the CelebA [23] and LFW [13] datasets, while the ones in b) come from the validation set of FFHQ [17]. Compared to FaceID-GAN, ConfigNet produces results with more consistent illumination and background.

Fig. 11: Change of hair colour from blond to black using ConfigNet (top) and synthetic data processed by CycleGAN (bottom). CycleGAN fails to preserve facial attibutes like expression and skin colour that should remain constant between the image pairs.

Table 3: Average controllability metrics for ConfigNet, CycleGAN and synthetic data. Ideally, $C_{pred}(I_+) = 1$, $C_{pred}(I_-) = 0$ and MD should be 0. The mean difference $C_{pred}(I_+) - C_{pred}(I_-)$ gives the dynamic range of a given attribute, the higher it is the more controllable the attribute.

Method	$C_{pred}(I_+) \uparrow$	$C_{pred}(I_-) \downarrow$	MD$\downarrow$	$C_{pred}(I_+) - C_{pred}(I_-) \uparrow$
base method	0.54	0.04	0.055	0.50
CycleGAN	0.52	0.01	0.080	0.51
synthetic data	0.46	0.01	0.051	0.45

that this is caused by the domain gap between the synthetic dataset and the CelebA dataset which was used to train the attribute predictor.

To compare ConfigNet to CycleGAN in terms of the photorealism of generated images, we generate an additional set of 10k synthetic images and pass them through CycleGAN. We then compute the Frechet Inception Distance (FID) between the converted synthetic images and images from the FFHQ validation set. The FID score for CycleGAN is 37.74, while the score for ConfigNet is 33.41 as reported in Section 4.

6.4 Implementation details

For the perceptual loss we use layers conv_1_2, conv_2_2, conv_3_4, conv_4_4 of VGG-19. We regularize all the discriminators with the R_1 gradient penalty described in [25]. In the image discriminators, we also use the style discriminator loss $\mathcal{L}_{style}$ described in HoloGAN [27], while in the generator we add the identity loss $\mathcal{L}_{identity}$ described in the same paper. While HoloGAN re-uses the discriminator features for identity loss, we use a separate network that has the same architecture as the image discriminators. We do so, because neither of our discriminators is trained to work with both real and synthetic data. We set the loss weights as follows: eye loss weight $\lambda_{eye} = 5$, domain adverserial loss weight $\lambda_{DA} = 5$, identity loss weight $\lambda_{identity} = 10$, gradient penalty loss weight $\lambda_{R_1} = 10$, perceptual loss weight in 1st stage $\lambda_{perc} = 0.00005$, perceptual loss weight in 2nd stage $\lambda_{perc} = 0.0005$. The adveserial losses on the images and style discriminator losses all have weight 1.

In the first training stage we sample $z \sim \mathcal{N}(0, \mathcal{I})$ and $r_R \sim \mathcal{U}(-r_{lim}, r_{lim})$, where r_R is the rotation sample for real data and r_{lim} is a pre-determined, per axis rotation limit. In all our experiments we set r_{lim} to be identical to the rotation limits used in synthetic data generation as described in the dataset section. In the second stage the E_R output corresponding to r_R is constrained to the range specified in r_{lim} by using a tanh activation and multiplying the output by r_{lim}.

Table 4 shows the architecture of the generator network G. In each AdaIN [14] input the latent vector z is processed by a 2-layer MLP. The volume rotation layer is the same as the one used in HoloGAN [27]. Table 5 shows the architecture of the image discriminators D_R, D_S. Following [27], most of the convolutional

Table 4: Architecture of the generator network G

Layer name	Kernel shape	Activation	Output shape	Normalisation
learned const input	-	-	$4 \times 4 \times 4 \times 512$	-
upsampling	-	-	$8 \times 8 \times 8 \times 512$	-
conv3d_1	$3 \times 3 \times 3$	LReLU	$8 \times 8 \times 8 \times 256$	AdaIN
upsampling	-	-	$16 \times 16 \times 16 \times 256$	-
conv3d_2	$3 \times 3 \times 3$	LReLU	$16 \times 16 \times 16 \times 128$	AdaIN
volume rotation	-	-	$16 \times 16 \times 16 \times 128$	-
conv3d_3	$3 \times 3 \times 3$	LReLU	$16 \times 16 \times 16 \times 64$	-
conv3d_4	$3 \times 3 \times 3$	LReLU	$16 \times 16 \times 16 \times 64$	-
reshape	-	-	$16 \times 16 \times (16 \cdot 64)$	-
conv2d_1	1×1	LReLU	$16 \times 16 \times 512$	-
conv2d_2	4×4	LReLU	$16 \times 16 \times 256$	AdaIN
upsampling	-	-	$32 \times 32 \times 256$	-
conv2d_3	4×4	LReLU	$32 \times 32 \times 64$	AdaIN
upsampling	-	-	$64 \times 64 \times 64$	-
conv2d_4	4×4	LReLU	$64 \times 64 \times 32$	AdaIN
upsampling	-	-	$128 \times 128 \times 32$	-
conv2d_5	4×4	LReLU	$128 \times 128 \times 32$	AdaIN
upsampling	-	-	$256 \times 256 \times 32$	-
conv2d_6	4×4	tanh	$256 \times 256 \times 3$	-

Table 5: Architecture of the image discriminator networks D_R, D_S

Layer name	Kernel shape, stride	Activation	Output shape	Normalisation
conv2d_1	$1 \times 1, 1$	-	$256 \times 256 \times 3$	-
conv2d_2	$3 \times 3, 2$	LReLU	$128 \times 128 \times 48$	Instance Norm
conv2d_3	$3 \times 3, 2$	LReLU	$64 \times 64 \times 96$	Instance Norm
conv2d_4	$3 \times 3, 2$	LReLU	$32 \times 32 \times 192$	Instance Norm
conv2d_5	$3 \times 3, 2$	LReLU	$16 \times 16 \times 384$	Instance Norm
conv2d_6	$3 \times 3, 2$	LReLU	$8 \times 8 \times 768$	Instance Norm
fully connected	49152	-	1	-

layers of the discriminator use instance normalization [40]. The latent GAN generator G_{lat} and discriminator share the same 3-layer MLP architecture.

The networks are optimized using Adam [19] with a learning rate of 4e-4. We perform the first stage of training for 50k iterations and then the second stage for 100k iterations. The latent GAN is also trained for 100k iterations. Following [16], in both the latent GAN and decoder G, we keep an exponential running mean of the weights during training and use those smoothed weights to generate all results.

6.5 Factorized latent space details

Table 6 shows the dimensionalities of all latent space factors z_i and corresponding synthetic data parameters θ_i. The dimensionalities of each z_i were chosen based

Table 6: Dimensionalities and descriptions of latent space factors

Factor name	dim θ_i	dim z_i	Description of θ_i
beard style	9	7	PCA coefficients
eyebrow style	44	7	PCA coefficients
expression	52	30	3D head model parameters $\in [0, 1]$
eye colour	6	3	one-hot encoding
eye rotation	3	2	rotation angles
hair colour	3	3	melanin, grayness, redness
hair style	18	8	PCA coefficients
head shape	53	30	3D head model parameters
illumination	50	20	PCA coefficients
lower eyelash style	3	2	one-hot encoding
texture	50	30	VAE latent space vector
upper eyelash style	3	2	one-hot encoding

on perceived complexity of the feature, for example we allocate more dimensions to expression than to hair colour. We are able to exert control over all the parameters with the exception of eyelash styles, which correspond to features that are too small at the image resolution we are working in. The expression parameters consist of the 51 expression blendshapes described in [3] and one additional dimension for the rotation of the jaw bone that leads to mouth opening.

6.6 Controllability metric details

The attribute predictor C_{pred} we use for the metrics is a MobilenetV2 [33] trained to predict 38 of CelebA's 40 attributes. The two attributes we do not predict are Wearing_Necklace and Wearing_Necktie as the required features are not present in our crops of CelebA images. For controllability metrics computed using the attribute predictor we use ConfigNet to drive 8 attributes, while we use all 38 attributes to compute the MD value. The 8 attributes we drive are chosen to be non-ambiguous and easy to verify by the user study participants.

For each evaluated face attribute v we set the corresponding $z_i = E_{S_i}(\theta_i)$, where θ_i is determined by manual inspection for both I_- and I_+. For example, for v smile we set the expression parameters that correspond to mouth corners going up for I_+, while for I_- we set parameters that correspond to mouth corners going down. Note that in this experiment we do not use the fine-grained control method described in Section 3.6, we instead set the entire θ_i with the chosen value.

6.7 Entanglement in synthetic dataset

While SynthFace allows for disentangling many face attributes, we have found that some of it's properties lead to entanglement. In SynthFace each texture is applied together with a corresponding displacement map, this leads to entanglement between texture and face shape. Examples of such entanglement are

noticeable in texture row of Figure 14. The eyebrow style row of the same figure shows changes in eyebrow height, which are entangled with the eyebrow raise expression. This is due to the varying vertical placement of eyebrows in SynthFace.

Another issue is that in SynthFace a neutral face can have an open mouth. Because of that, applying a neutral expression does not always lead to the mouth closing, this issue is visible in top row of Figure 7. In those cases, the mouths can still be closed by setting the value of the mouth opening expression to negative. The same issue applies to other expressions to a smaller degree.

6.8 Additional figures

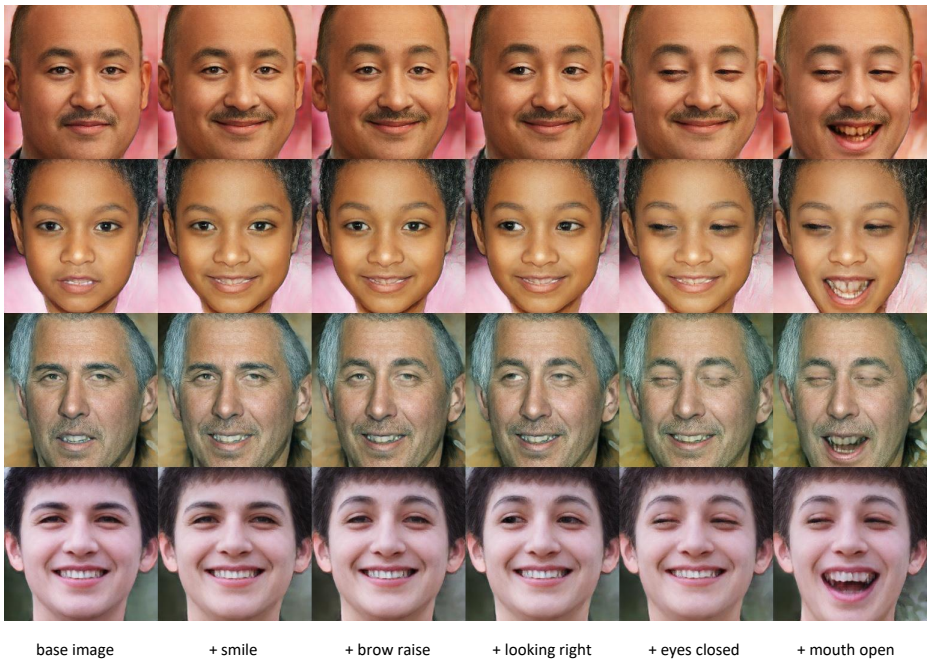

Fig. 12: Generation of combinations of facial expressions. The left column shows the base image while each subsequent column adds another expression to the image. With the exception of looking right each expression is added using the fine-grained control method described in Section 3.6.

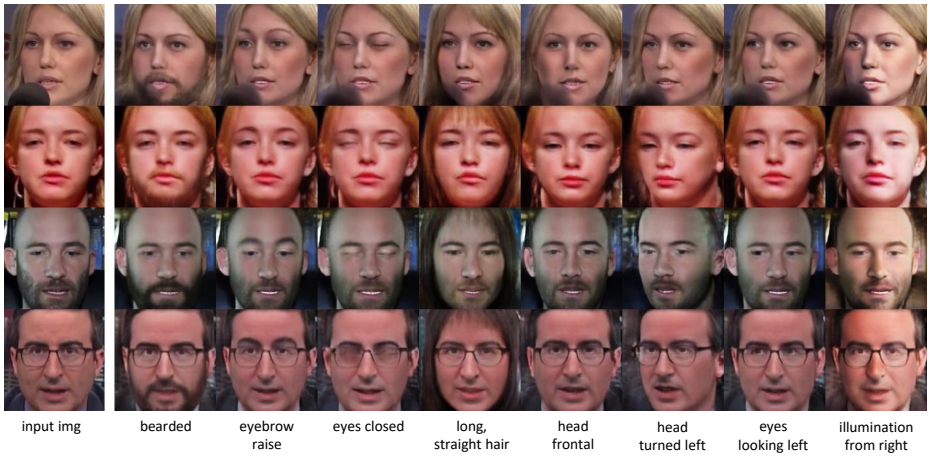

Fig. 13: Modification of various attributes of faces from the 300-VW dataset [32] using ConfigNet fine-tuned on the input image.

Algorithm 1: Fine grained control. The algorithm uses projected gradient descent to estimate the synthetic data rendering parameter $\tilde{\theta}_i$ that maps to the supplied z_i. $\tilde{\theta}_i$ can then be modified to obtain a new $z_i = E_S(\tilde{\theta}_i)$. Since $\tilde{\theta}_i$ is a parameter of a computer graphics pipeline, it is semantically meaningful and its individual components can be modified to achieve fine-grained control.

Input: latent space factor z_i, number of iterations n, learning rate γ
Result: modified latent space factor z_i
iter = 0;
$\tilde{\theta}_i = [0, 0, \ldots, 0]$;
while $iter < n$ **do**
  $L = |z_i - E_{S_i}(\tilde{\theta}_i)|^2$;
  $\tilde{\theta}_i = \tilde{\theta}_i - \gamma \nabla L$;
  clip $\tilde{\theta}_i$ to the range of valid values of θ_i;
  iter++;
end
apply desired modification to $\tilde{\theta}_i$;
$z_i = E_{S_i}(\tilde{\theta}_i)$;

Fig. 14: Random variations of the input factors of ConfigNet shown on a single base image. In each row a single latent space factor $z_i = E_S(\theta_i)$ is modified, with each row showing a different, randomly selected θ_i. Note the change of appearance when modifying head shape, discussion in Section 4.4.

Fig. 15: Results of sampling the latent space $z = E_R(I_R)$, where I_R is randomly selected from FFHQ [17].

(a) Model trained with the eye gaze loss

(b) Model trained without the eye gaze loss

Fig. 16: Comparison of control over eye gaze direction for a model trained with the eye gaze loss $\mathcal{L}_{eye}$ and a model trained without it. Notice that a) achieves a wider range of eye motion, which is most visible in the first and last column of each row.

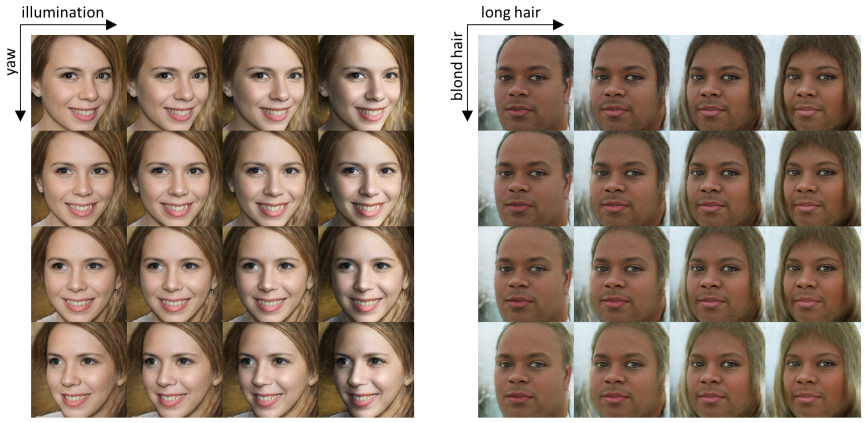

Fig. 17: Interpolation of two face attribute at a time in images sampled from the FFHQ [17] dataset. Note that the illumination direction stays constant as the head rotates.

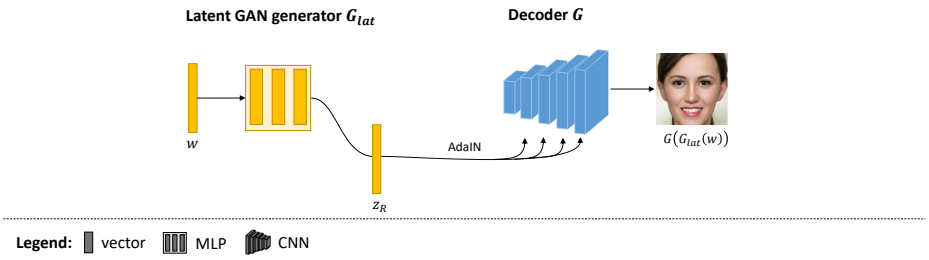

Fig. 18: Outline of the method at inference time, when the latent GAN (described in Section 3.5) is used to sample the latent space $z_r = G_{lat}(w)$, where $w \sim \mathcal{N}(0, \boldsymbol{I})$. Keep in mind that z_r can still be modified using any of the methods described in the paper to achieve control over the output image.