CONFIG: Controllable Neural Face Image Generation

Marek Kowalski, Stephan J. Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton
Microsoft

Abstract. Our ability to sample realistic natural images, particularly faces, has advanced by leaps and bounds in recent years, yet our ability to exert fine-tuned control over the generative process has lagged behind. If this new technology is to find practical uses, we need to achieve a level of control over generative networks which, without sacrificing realism, is on par with that seen in computer graphics and character animation. To this end we propose ConfigNet, a neural face model that allows for controlling individual aspects of output images in semantically meaningful ways and that is a significant step on the path towards finely-controllable neural rendering. ConfigNet is trained on real face images as well as synthetic face renders. Our novel method uses synthetic data to factorize the latent space into elements that correspond to the inputs of a traditional rendering pipeline, separating aspects such as head pose, facial expression, hair style, illumination, and many others which are very hard to annotate in real data. The real images, which are presented to the network without labels, extend the variety of the generated images and encourage realism. Finally, we propose an evaluation criterion using an attribute detection network combined with a user study and demonstrate state-of-the-art individual control over attributes in the output images.

Keywords: neural rendering; face image manipulation; GAN;

1 Introduction

Recent advances in generative adversarial networks (GANs) [14,16] have enabled the production of realistic high resolution images of smooth organic objects such as faces. Generating photorealistic human bodies, and faces in particular, with traditional rendering pipelines is notoriously difficult [22], requiring hand-crafted 3D assets. However, once these assets have been generated we can render the face from any direction and in any pose. In contrast, GANs can be used to easily generate realistic head and face images without the need to author expensive 3D assets, by training on curated datasets of 2D images of real human faces. However, it is difficult to enable meaningful control over this generation without detailed hand labelling of the dataset. Even when conditional models are trained
Fig. 1: ConfigNet learns a factorized latent space, where each part corresponds to a different facial attribute. The first column shows images produced by ConfigNet for certain points in the latent space. The remaining columns show changes to various parts of the latent space vectors, where we can generate attribute combinations outside the distribution of the training set like children or women with facial hair.

Fig. 2: ConfigNet has two encoders \( E_R \) and \( E_S \) that encode real face images \( I_R \) and the parameters \( \theta \) of synthetic face images \( I_S \). The encoders output latent space vectors \( z_R, z_S \). The shared decoder, \( G \), generates both real and synthetic images. A domain discriminator \( D_{DA} \) ensures the latent distributions generated by \( E_R \) and \( E_S \) are similar.
with detailed labels, they struggle to generalize to out-of-distribution combinations of control parameters such as children with extensive facial hair or young people with gray hair. In order for GAN based rendering techniques to replace traditional rendering pipelines they must enable a greater level of control.

In this paper we present ConfigNet, the first method to enable control of GAN outputs using the same methods as traditional graphics pipelines. The key idea behind ConfigNet is to train the generative model on both real and synthetically generated face images. Since the synthetic images were generated with a traditional graphics pipeline, the renderer parameters for those images are readily available. We use those known correspondences to train a generative model that uses the same input parametrization as the graphics pipeline used to generate the synthetic data. This allows for independent control of various face aspects including: head pose, hair style, facial hair style, expression and illumination. By simultaneously training the model on unlabelled face images, it learns to generate photorealistic looking faces, while enabling full control over these outputs. Figure 1 shows example results produced by ConfigNet.

ConfigNet can be used to both sample novel images and to embed existing ones, which can then be manipulated. The ability to embed face images and manipulate them sets ConfigNet apart from traditional graphics pipelines, which would require person-specific 3D assets to achieve similar results. The process of generating such person-specific assets usually requires a 3D capture rig and significant artist effort. Our method also required 3D assets to render the synthetic data, however, it allows for generating realistic images of specific people without requiring person-specific assets. Moreover, the realism of the images generated using our method is higher than that of the synthetic data we use.

The use of a parametrization derived from a traditional graphics pipeline makes ConfigNet easy to use for people familiar with digital character animation. For example, facial expressions are controlled with blendshapes with values in (0, 1), head pose is controlled with Euler angles and illumination can be set using an environment map.

Our main contributions are:

1. The first attempt at neural rendering where the input parametrization is fully based on a traditional computer graphics pipeline. Comparable existing methods only make small parts of their input space controllable in this way.
2. ConfigNet, a novel method for placing real and synthetic data into a single factorized and disentangled latent space.
3. A method for using ConfigNet to modify existing face images in a fine-grained way that allows for changing parts of the latent space factors meaningfully.
4. Experiments showing method generating realistic face images with attribute combinations that are not present in the real images of the training set. For example, a face of a child with extensive facial hair.
2 Related work

Image generation driven by synthetic data One of the most common uses of synthetic data in image generation is the “synthetic to real” scenario, where the goal is to generate realistic images that belong to a target domain based on synthetic images, effectively increasing their realism. The methods that tackle this problem \[30,7,39\] usually use a neural network with an adversarial and semantic loss to push a synthetic image closer to the real domain. While those type of methods can generate realistic images that are controllable through synthetic data, the editing of existing images is difficult as it would require fitting the underlying 3D model to an existing image.

PuppetGAN \[35\], on the other hand, is a method designed to edit existing images using synthetic data of the same class of objects. It uses two encoder-decoder pairs, one for real and one for synthetic images, which have a common latent space, part of which is designated for an attribute of interest that can be controlled. An image can be edited by encoding it with the real-data encoder, then swapping the attribute of interest part of the latent space with one encoded from a synthetic image and finally decoding with the real-data decoder. Due to the use of separate decoders for real and synthetic data PuppetGAN struggles to decode images where the attribute of interest is outside of the range seen in real data. The method performs well for a single attribute. In contrast, ConfigNet demonstrates disentanglement of multiple face attributes as well as generation of attribute combinations that do not exist in the real training data.

Disentangled representation learning Supervised disentanglement methods try to learn a factorised representation, parts of which correspond to some semantically meaningful aspects of the generated images, based on labelled data in the target domain (as opposed to the synthetic data domain). The major limitation of these methods \[20,38,25\] is that they are only able to disentangle factors of variations that are labelled in the training set. For human faces, labels are easily obtainable for some attributes, such as identity, but the task becomes more difficult with attributes like illumination and almost impossible with attributes like hair style. This labelling problem also becomes more difficult as the required fidelity of the labels increases (e.g. smile intensity).

Unsupervised disentanglement methods share the above goal but do not require labelled data. Most methods in this family, such as $\beta$-VAE \[11\], InfoGAN \[6\], ID-GAN \[18\], place constraints on the latent space that lead to disentanglement. The fundamental problem with those approaches is that there is no control over what factors get disentangled and which part of the latent space corresponds to a given factor of variation. HoloGAN \[23\] separates the 3D rotation of the object in the image from variation in its shape and appearance. ConfigNet borrows the generator architecture of HoloGAN, while disentangling many additional factors of variation and allowing existing images to be edited.

Face video re-enactment Face video re-enactment methods aim to produce a video of a certain person’s articulated face that is driven by a second video of the same or a different person. The methods in \[32,38,36\] have achieved some of the most impressive face manipulation results seen to date. Face2Face \[32\] fits
a 3D face model and illumination parameters to a video of a person and then re-renders the sequence with modified expression parameters that are obtained from a different sequence. This approach potentially allows for modifying any aspects of the rendered face that can be modelled, rendered and fitted to the input video. In practice, due to limitations of existing 3D face models and fitting methods, this approach cannot modify complex face attributes like hair style or attributes that require modelling of the whole head, like head pose.

Zakharov et al. [38] propose a video re-enactment method where the images are generated by a neural network driven by face landmarks from a different video sequence. The method produces impressive results given only a small number of target frames. X2Face [36] uses one neural network to resample the source image into a standard reference frame and a second network that resamples this standardized image into a different head pose or facial expression, which can be driven by images or audio signal. Interestingly, since the method resamples the standardized frame to produce the output, any modifications to the standardized frame will be present in the output. One can, for example, manually draw a shape on the standardized frame and it will appear in the outputs. While these two methods produce convincing results, the controllability is limited to head pose and expression. Another downside is that both methods require video data for training and there are currently no publicly available face video datasets at resolutions comparable to image datasets like CelebA-HQ [14] or FFHQ [15].

3 Method

The key concept behind the proposed method is to factorize the latent space into parts that correspond to separate and clearly-defined attributes of face images. This requires labels that fully explain the image content, which require laborious annotation for real data, but straightforward for synthetic data. We thus propose a generative model trained in a semi-supervised way, with labels that are known for synthetic data only. Figure 2 outlines the proposed architecture.

**Overview** Our approach is to treat the synthetic images $I_S$ and real images $I_R$ as two different subsets of a larger set of all possible face images. Hence, the proposed method consists of a decoder $G$ and two encoders $E_R$ and $E_S$ that embed real and synthetic data into a common factorized latent space $z$ (Section 3.1). We will refer to $z$ predicted by $E_R$ and $E_S$ as $z_R$ and $z_S$ respectively. While the real data is supplied to the encoder as images $I_R \in I_R$, the synthetic data is supplied as vectors $\theta \in \mathbb{R}^m$ that fully describe the content of the corresponding image $I_S \in I_S$. To increase the realism of the generated images we employ two discriminator networks $D_R$ and $D_S$ for real and synthetic data respectively.

We assume that the synthetic data is a reasonable approximation of the real data so that $I_S \cap I_R \neq \emptyset$. Hence, it is desirable for $E_S(\theta)$ and $E_R(I_R)$, where $\theta$ is the space of all $\theta$, to also be overlapping. To do so, we introduce a domain adversarial loss [33] on $z$, that forces $z_R$ and $z_S$ to be close to each other. In Section 4.2 we show that this loss is crucial for the method’s ability to control the attributes of the output images.
To accurately reproduce and modify existing images we employ one-shot learning (Section 3.4) that improves reconstruction accuracy compared to embedding using $E_R$. To enable the sampling of novel images we train a latent GAN that generates samples of $z$ (Section 3.5). Finally, we propose a method for modifying attributes of existing images in a fine-grained way that allows for changing parts of individual factors of $z$ meaningfully (Section 3.6).

3.1 Factorized latent space

Each synthetic data sample $\theta$ is factorised into $k$ parts $\theta_1$ to $\theta_k$, such that:

$$\theta \in \mathbb{R}^m = \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \ldots \times \mathbb{R}^{m_k}.$$  

Each $\theta_i$ corresponds to semantically meaningful input of the graphics pipeline used to generate $I_S$. Examples of such inputs are: facial expression, facial hair parameters, head shape, environment map, etc. The synthetic data encoder $E_S$ maps each $\theta_i$ to $z_i$, a part of $z$, which thus factorizes $z$ into $k$ parts.

The factorized latent space is a key feature of ConfigNet that allows for easy modification of various aspects of the generated images. For example, one might encode a real image into $z$ using $E_R$ and then change the illumination by swapping out the part of $z$ that corresponds to illumination. Note that the part of $z$ that is swapped in might come from $\theta_i$, which is semantically meaningful, or it may come from a different real face image encoded by $E_R$.

3.2 Loss functions

To ensure that the output image $G(z)$ is close to the corresponding ground truth image $I_{GT}$, we use the perceptual loss $L_{\text{perc}}$ which is the MSE between the activations of a pre-trained neural network computed on $G(z)$ and $I_{GT}$. We use VGG-19 [31] trained on ImageNet [26] as the pre-trained network. We also experimented with using VGGFace [24] as base for the perceptual loss, but we didn’t see noticeable improvement.

While the perceptual loss retains the overall content of the image well, it struggles to preserve some small scale features. Because of that we use an additional loss with the goal of preserving the eye gaze direction:

$$L_{\text{eye}} = w_M \sum M \circ (I_{GT} - G(z_s)) \text{ with } w_M = (1 + |M|_1)^{-1},$$

where $M$ is a pixel-wise binary mask that denotes the iris, only available for $I_S$. Thanks to the accurate ground truth segmentation that comes with the synthetic data, similar losses could be added for any part of the face if necessary.

We train the adversarial blocks with the non-saturating GAN loss [8]:

$$L_{GAN_D}(D, x, y) = \log D(x) + \log(1 - D(y)),$$

$$L_{GAN_G}(D, y) = \log(D(y)),$$

where $L_{GAN_D}$ is used for the discriminator and $L_{GAN_G}$ is used for the generator, $D$ is the discriminator, $x$ is a real sample and $y$ is the generated sample.
3.3 Two-stage Training procedure

**First stage:** we train all the sub-networks except $E_R$, sampling $z_R \sim \mathcal{N}(0, I)$ as there is no encoder for real data at this stage. At this stage $E_S$ and $G$ are trained with the following loss:

$$
L_1 = \mathcal{L}_{GAN}(D_R, G(z_R)) + \mathcal{L}_{GAN}(D_{DA}, z_S) + \mathcal{L}_{GAN}(D_S, G(z_S)) + \lambda_{eye}\mathcal{L}_{eye} + \lambda_{perc}\mathcal{L}_{perc}(G(z_S), I_S),
$$

(5)

where $z_S = E_S(\theta)$ and $\lambda$ are the weights assigned to the corresponding losses. The domain discriminator $D_{DA}$ acts on $E_S$ to bring the distribution of its outputs closer to $\mathcal{N}(0, I)$ and so $E_S$ effectively maps the distribution of each $\theta_i$ to $\mathcal{N}(0, I)$.

**Second stage:** we add the real data encoder $E_R$ so that $z_R = E_R(I_R)$. The loss used for training $E_S$ and $G$ is then:

$$
L_2 = L_1 + \lambda_{perc}\mathcal{L}_{perc}(G(z_R), I_R) + \log(1 - D_{DA}(z_R)),
$$

(6)

where the goal of $\log(1 - D_{DA}(z_R))$ is to bring the output distribution of $E_R$ closer to that of $E_S$. In the second stage we increase the weight of $\lambda_{perc}$, in the first stage it is set to a lower value as otherwise total loss for synthetic data would overpower that for real data. In the second stage both real and synthetic data use the perceptual loss and we increase its weight. Our experiments show that this two-stage training improves controllability and image quality.

3.4 One-shot learning by fine-tuning

Our architecture allows for embedding face images into $z$ using the real data encoder $E_R$, individual factors $z_i$ can then be modified to modify the corresponding output image. While experimenting with this architecture we have found that while $G(E_R(I_R))$ is usually similar to $I_R$ as a whole image, there is often an identity gap between the face in $I_R$ and in the generated image. A similar finding was made in [38], where the author proposed to decrease the identity gap by fine-tuning the generator on the images of a given person.

Similarly we fine-tune our generator on $I_R$ by minimizing the following loss:

$$
L_{ft} = \mathcal{L}_{GAN}(D_R, I_R, G(z_R)) + \log(1 - D_{DA}(z_R)) + \lambda_{perc}\{\mathcal{L}_{perc}(G(z_R), I_R) + \mathcal{L}_{face}(G(z_R), I_R)\},
$$

(7)

where $\mathcal{L}_{face}$ is a perceptual loss with VGGFace [24] as the pre-trained network. We optimize over the weights of $G$ as well as $z_R$ which is initialized with $E_R(I_R)$. The addition of a $\mathcal{L}_{face}$ improves the perceptual quality of the generated face images. We believe that this improvement is visible here, but not in the main training phase, as fine-tuning lacks the regularization provided by training on a large number of images and can easily “fool” the single perceptual loss.
3.5 Sampling of \( z \)

While the proposed method allows for embedding existing face images into the latent space, sometimes it might be desirable to sample the latent space itself. Samples of the latent space can be used to generate novel images or to sample individual factor \( z_i \). The sampled \( z_i \) can then be used to generate additional variations of an existing image that was embedded in \( z \). To do this we use a latent GAN \([1]\). The latent GAN is trained to map between its input \( w \sim \mathcal{N}(0, I) \) and the latent space \( z \). This simple approach allows for sampling the latent space without the constraints on \( z \) imposed by VAEs that lead to reduced quality. The latent GAN is trained with the GAN losses described above, both the discriminator and generator \( G_{lat} \) are 3-layer MLPs.

3.6 Fine-grained control

Given an existing face image embedded into \( z \), we can easily swap any part, \( z_i \), of its embedding with one that is obtained from \( E_S \) or \( E_R \). However, sometimes we might want a finer level of control and only modify a single aspect of \( z_i \) while leaving the rest the same. If \( z_i \) is a face expression, its single aspect might be the intensity of smile, if \( z_i \) is illumination, the brightness might be one aspect. These aspects are controlled by individual elements of the corresponding \( \theta_i \) vector. However \( \theta_i \) is unknown if the \( z \) was generated by \( E_R \) or \( G_{lat} \).

For this reason, we use an approximation \( \tilde{\theta}_i \) obtained by solving the minimization problem \( \min_{\tilde{\theta}_i} |z_i - E_S(\tilde{\theta}_i)|^2 \) with gradient descent, where \( E_S \) is the part of \( E_S \) that corresponds to \( \theta_i \). We incorporate constraints on \( \theta_i \) into the optimization algorithm. For example, our expression parameters lie in the convex set \([0, 1]\) and we use projected gradient descent to incorporate the constraint into the minimization algorithm. Given \( \tilde{\theta}_i \), e.g. a face expression vector, we can modify the part of the vector responsible for an individual expression and use \( E_S \) to obtain a new latent code \( z_i \) that generates images where only this individual expression is modified. We use this approach to manipulate individual expressions in Figure 1 (smile and eyes closed) and Figure 7 (mouth opening).

3.7 Implementation

The architecture of the decoder \( G \) is based on the generator used in HoloGAN \([23]\), explained in supplementary. We choose this particular architecture as it decouples object rotation from the latent space and it allows for specifying the object rotation with any parametrization (Euler angles or quaternions). This lets us obtain the poses of the heads in \( I_S \) in ConfigNet parametrization and supply head pose directly, without requiring an encoder.

The remaining \( k - 1 \) parts of \( \theta \) are encoded with separate multi layer perceptrons (MLPs) \( E_{S_i} \), each of which consists of 2 layers with number of hidden units equal to the dimensionality of the corresponding \( \theta_i \). The real image encoder \( E_R \) is a ResNet-50 \([9]\) pre-trained on ImageNet \([27]\). The domain discriminator \( D_{DA} \)
Fig. 3: Images from SynthFace dataset, note the domain gap to real images.

is a 4-layer MLP. The two image discriminators $D_R$ and $D_S$ share the same basic convolutional architecture. The supplementary material contains all network details and we plan to release the source code of ConfigNet upon publication.

4 Experiments

Datasets We use the FFHQ [15] (60k images, 1Mpix each), and SynthFace (30k images, 1Mpix each) datasets as a source of real and synthetic training images. We align the face images from all datasets to a standard reference frame using landmarks from OpenFace [4,37,3] and reduce the resolution to 256x256 pixels.

Our experiments use the 10k images in the validation set of FFHQ to evaluate ConfigNet. The SynthFace dataset was generated using the original method of [2] and setting rotation limits for yaw and pitch to $\pm 30^\circ$ and $\pm 10^\circ$ to cover the typical range of poses in face images. For SynthFace, $\theta$ has $m = 304$ dimensions, while $z$ has $n = 145$ dimensions, and is divided into $k = 12$ factors. The supplementary material provides the dimensionality of each factor in $\theta$ and $z$ in SynthFace and Figure 3 shows sample images.

4.1 Evaluation of ConfigNet

Our experiments evaluate ConfigNet key features: photorealism and control.

Photorealism Figure 4 shows samples generated by the latent GAN (where $E_R, G$ were trained using the two stage-procedure of Section 3.3) and a standard GAN model trained only with the first-stage procedure. We observe a large improvement in photorealism when the second stage of training is added. We believe that the low-quality images produced by the standard GAN are caused by the constraint $z \sim \mathcal{N}(0, I)$, which is relaxed in our second-stage training thus allowing real and synthetic data to co-exist in the same space.

We quantitatively measure the photorealism and coverage of the generated images using the Frechet Inception Distance (FID) [10] in Table 1. The latent GAN achieves scores that are close to those produced by sampling $z$ through $E_R$, which is the upper limit of its performance. Training only the first stage and sampling $z \sim \mathcal{N}(0, I)$ results in poorer metrics. As expected, the raw synthetic images give the worst result. To further evaluate how much of the photorealism of the generated data is lost due to training on both real and synthetic data, we train ConfigNet without synthetic data and the losses that require its presence. We find that the resulting FID and IS are very close to those produced by our standard training. This suggests that the photorealism of the results might be
Fig. 4: Left: $G(z)$ trained using the two-stage method, where $z$ is sampled from latent GAN. Right: $G(z)$ trained using the first stage only, where $z$ is sampled from prior. Note the large improvement in quality when second stage is added.

Table 1: FID score for FFHQ, SynthFace, and images obtained with our decoder $G$ and latent codes from the real-image encoder $E_R$ and latent GAN $G_{lat}$.

| Method                              | FID ↓ |
|-------------------------------------|-------|
| $G(E_R(I_R))$                       | 33.41 |
| synthetic data $\mathcal{I}_S$      | 52.19 |
| $G(E_R(I_R))$ without $\mathcal{I}_S$| 33.49 |
| $G(G_{lat}(w)), w \sim \mathcal{N}(0, (I))$ | 39.76 |
| $G(z), z \sim \mathcal{N}(0, (I))$ no 2nd stage | 43.05 |

limited by our network architecture rather than by the use of synthetic data. We speculate that using a more powerful $G$ and $D_R$ may lead to improved results.

**Controllability** We evaluate ConfigNet’s controllability analysing how changing a specific attribute (e.g., hair colour) changes the output image: with perfect control, the output image should only change with respect to that attribute.

Figure 1 and 6 show controllability qualitatively. Figure 1 shows that the generator is able to modify individual attributes of faces embedded in its latent space, while Figure 6c shows that each attribute can take many different values while only influencing certain aspects of the produced image. The second column of Figure 6 shows that we are able to set facial hair to faces of children and women, demonstrating that the generator is not constrained by the distribution of the real training data. The supplementary includes additional results of face attribute manipulation and interpolation, including a video.

To evaluate if ConfigNet offers this ideal level of control quantitatively, we propose the following experiment: We take a random image $I_R$ from the FFHQ validation set, encode it into latent space $z = E_R(I_R)$ and then use the synthetic encoder $E_S$ to modify the latent factor $z_i$ that corresponds to a given attribute $v$. For each attribute $v$ we output two images: $I_+$ where the attribute is set to a certain value $v_+$ (e.g., blond hair) and another $I_-$ with the attribute takes a semantically opposite value $v_-$. (e.g., black hair$^1$). This gives us image pairs

---

$^1$ We choose the values of $v_+$ and $v_-$ by manual inspection, as explained in supplementary material.
(a) Evaluation of controllability and disentanglement with an attribute predictor.

(b) Evaluation of controllability and disentanglement with a user study.

Fig. 5: Evaluation of control and disentanglement of ConfigNet. Blue and orange bars show the predicted values of given attribute for images with that attribute \((I_+, \text{ higher is better})\) and images with an opposite attribute \((I_-, \text{ lower is better})\). The gray bars measure differences of other attributes (MD and \(C_{\text{diff}}\), lower is better).

\((I_+, I_-)\) that should be identical except for the chosen attribute \(v\), where they should differ. We measure how and where these images differ with an attribute predictor and a user study.

We train an attribute predictor \(C_{\text{pred}}\) on CelebA [19] to predict 38 face attributes and use it with 1000 FFHQ validation images to estimate 1) if \(v_+\) is present in each set of images pairs \((I_+, I_-)\) and 2) if the other face attributes change. Ideally, \(C_{\text{pred}}(I_+) = 1, C_{\text{pred}}(I_-) = 0\) and the Mean Absolute Difference (MD) for other face attributes should be 0. Figure 5a shows how \(C_{\text{pred}}(I_+) \gg C_{\text{pred}}(I_-)\) while the MD of other attributes is close to 0. The best controllability is achieved for the mouth opening and smile attributes, with \(C_{\text{pred}}(I_+)\) approaching the ideal value of 1, while the poorest results are achieved for the gray hair attribute. We believe those large differences are caused by bias in CelebA, where certain attributes are not distributed evenly across age (for example gray hair) or gender (for example moustache).

Our user study \(C_{\text{user}}\) follows a similar evaluation protocol: 59 users evaluated the presence of \(v_+\) in a total of 1771 images pairs \(I_+\) and \(I_-\) on a 5-level scale and gave a score \(C_{\text{diff}}\) that measures whether, ignoring \(v\), the images depict the same person. Figure 5b shows the results of the controllability and disentanglement metrics for the user study: users evaluate the controllability of the given attribute higher than the feature predictor \(C_{\text{pred}}\), with \(C_{\text{user}}(I_+) > C_{\text{pred}}(I_+)\) and \(C_{\text{user}}(I_+) - C_{\text{user}}(I_-) > C_{\text{pred}}(I_+) - C_{\text{pred}}(I_-)\) for all features except mouth.
Table 2: Average controllability metrics for different variants of ConfigNet. $D_{DA}$ denotes the domain discriminator. Ideally, $C_{\text{pred}}(I_+) = 1$, $C_{\text{pred}}(I_-) = 0$ and MD should be 0. The mean difference $C_{\text{pred}}(I_+) - C_{\text{pred}}(I_-)$ gives the dynamic range of a given attribute, the higher it is the more controllable the attribute.

| Method                | $C_{\text{pred}}(I_+)$ ↑ | $C_{\text{pred}}(I_-)$ ↓ | MD  | $C_{\text{pred}}(I_+) - C_{\text{pred}}(I_-)$ ↑ |
|-----------------------|--------------------------|--------------------------|-----|-----------------------------------------|
| base method           | 0.54                     | 0.04                     | 0.06 | 0.50                                    |
| with fine-tuning      | 0.52                     | 0.05                     | 0.05 | 0.47                                    |
| without $D_{DA}$      | 0.39                     | 0.19                     | 0.03 | 0.20                                    |
| without 1st stage     | 0.43                     | 0.14                     | 0.04 | 0.29                                    |

open, while the score $C_{\text{diff}}$ measuring whether $I_+$ and $I_-$ show the same person has low values indicating that features other than $v_+$ remain close to constant. This results support the result of the feature predictor and show a similar performance for different attributes because user judgements do not suffer from the bias of the attribute predictor trained on CelebA.

4.2 Ablation study

We evaluate the importance of two stage training and the domain discriminator $D_{DA}$ by training the neural network without them. Table 2 shows how each of those procedures contributes to controllability of ConfigNet. Compared to the base method, $C_{\text{pred}}(I_+) - C_{\text{pred}}(I_-)$ decreases by 60% when the domain adversarial loss is removed and by 42% when the first stage training is removed. Quantitatively, the mean absolute difference of the non-altered attributes, MD, is slightly larger for the base method. While this might seem a degradation caused by two stage training and the domain discriminator, we attribute this change to the reduced capability of the network to modify the output image, and hence to lower difference in other attributes as well.

One worry with fine-tuning\(^2\) on a single image is that it will change the decoder in a way that negatively affects controllability of the output image. Our experiments show that fine tuning leads to a 6% reduction in $C_{\text{pred}}(I_+) - C_{\text{pred}}(I_-)$ and no increase of MD, which leads us to believe that the controllability of the fine-tuned generator is not significantly affected. Figure 6 qualitatively shows the effects of fine-tuning compared to embedding using $E_R$.

4.3 Comparison to state of the art

To the best of our knowledge, the only method that is directly comparable to ConfigNet is PuppetGAN \(^3\). For comparison we use a figure from \(^3\) that shows control over the degree of mouth opening on frames from several videos from the 300-VW dataset \(^2\). To generate the figure, the authors of PuppetGAN trained separate models on each of the videos and then demonstrated the ability to change the degree of mouth opening in the frames of the same video.

\(^2\) In all fine-tuning experiments we ran the fine-tuning procedure for 50 iterations.
Fig. 6: Effects of fine tuning and attribute variety. The first 3 columns show the input image, the results of the encoder embedding and fine tuning. The other columns show different facial attributes controllable modifying $E_S(\theta_i)$.

To generate similar results we use a model trained on FFHQ and fine-tune it on the input frame using the method described in Section 3.4. We then use the fine-grained control method (Section 3.6) to change only the degree of mouth opening. The results of this comparison are shown in Figure 7. At a certain level of mouth opening PuppetGAN saturates and is not able to open the mouth more widely, ConfigNet does so, while retaining a similar level of quality and disentanglement. Both methods fail to close the mouth fully for some of the input images. We believe that in case of ConfigNet this is an issue with the disentanglement of the synthetic training set itself, we give further details and describe a solution in supplementary materials. It is also worth noting that PuppetGAN uses hundreds of training images of a specific person, while ConfigNet requires only a single frame and it is able to modify many additional attributes.

4.4 Failure modes

One of the key issues we have identified is that the $z_i$ that corresponds to head shape is often separated for real and synthetic data. For example, changing the head shape of a real image embedded into $z$ using $E_S(\theta_i)$ results in the face appearing closer to the synthetic image space and some of its features being lost, see Figure 8a for an example. This separation is placed in the head shape space very consistently, we believe this is because head shape affects the whole image in a significant way, so it’s easy for the generator to “hide” the difference between real and synthetic images there.

Another issue is that SynthFace does not model glasses, which leads to ConfigNet hiding the representation of glasses in unrelated face attributes, most commonly texture, head and eyebrow shape, as shown in Figure 8b. Lastly, we have found that when $I_R$ has a head pose that is out of the rotation range of SynthFace, the encoder $E_R$ hides the rotation in other parts of $z$, as shown in
Fig. 7: Comparison between ConfigNet (left) and PuppetGAN (right). Top row shows the input frame and the left column the desired level of mouth opening for each row. To facilitate visual comparison, ConfigNet results are cropped to match PuppetGAN.

Fig. 8: Failure modes. a) Changing head shape to one obtained from $\theta$ moves the appearance of the image closer to synthetic data. b) Change of $z_i$ corresponding to texture changes style of glasses. c) Frontal image generated from an image $I_R$ with pose outside the supported range.

Figure 8c. We believe this is a result of constraining the rotation output of $E_R$ to the range seen in SynthFace (details in supplementary). Generating a synthetic dataset with a wider rotation range would likely alleviate this issue.

5 Conclusions

We have presented ConfigNet, a novel face image synthesis method that allows for controlling the output images to an unprecedented degree. Crucially, we show the ability to generate realistic face images with attribute combinations that are outside the distribution of the real training set. This unique ability brings neural rendering closer to traditional rendering pipelines in terms of flexibility.
An open question is how to handle aspects of real face images not present in synthetic data. Adding additional variables in the latent space to model these aspects only for real is an investigation that we leave to future work.

In the short term, we believe that ConfigNet could be used to enrich existing datasets with samples that are outside of their data distribution or be applied to telepresence and character animation. In the long term, we hope that similar methods will replace traditional rendering pipelines and allow for controllable, realistic and person-specific face rendering.

Acknowledgments The authors would like to thank Nate Kushman for helpful discussions and suggestions.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. arXiv preprint arXiv:1707.02392 (2017)
2. Authors: Synthface: a high quality synthetic face image dataset. in submission, supplied as additional material synthface.pdf
3. Baltrusaitis, T., Robinson, P., Morency, L.P.: Constrained local neural fields for robust facial landmark detection in the wild. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 354–361 (2013)
4. Baltrusaitis, T., Zadeh, A., Lim, Y.C., Morency, L.P.: Openface 2.0: Facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). pp. 59–66. IEEE (2018)
5. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
6. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in neural information processing systems. pp. 2172–2180 (2016)
7. Gecer, B., Bhattarai, B., Kittler, J., Kim, T.K.: Semi-supervised adversarial learning to generate photorealistic face images of new identities from 3d morphable model. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 217–234 (2018)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in neural information processing systems. pp. 6626–6637 (2017)
11. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. Icml 2(5), 6 (2017)
12. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
13. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
14. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
15. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
16. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. arXiv preprint arXiv:1912.04958 (2019)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Lee, W., Kim, D., Hong, S., Lee, H.: High-fidelity synthesis with disentangled representation. arXiv preprint arXiv:2001.04296 (2020)
19. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
20. Mathieu, M.F., Zhao, J.J., Zhao, J., Ramesh, A., Sprechmann, P., LeCun, Y.: Disentangling factors of variation in deep representation using adversarial training. In: Advances in neural information processing systems. pp. 5040–5048 (2016)
21. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? arXiv preprint arXiv:1801.04406 (2018)
22. Mori, M., et al.: The uncanny valley. Energy 7(4), 33–35 (1970)
23. Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.L.: Hologan: Unsupervised learning of 3d representations from natural images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7588–7597 (2019)
24. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition (2015)
25. Qian, S., Lin, K.Y., Wu, W., Liu, Y., Wang, Q., Shen, F., Qian, C., He, R.: Make a face: Towards arbitrary high fidelity face manipulation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 10033–10042 (2019)
26. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision 115(3), 211–252 (2015)
27. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision 115(3), 211–252 (2015)
28. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 397–403 (2013)
29. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
30. Shirvastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2107–2116 (2017)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
32. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of rgb videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2387–2395 (2016)
33. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7167–7176 (2017)
34. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
35. Usman, B., Dufour, N., Saenko, K., Bregler, C.: Puppetgan: Cross-domain image manipulation by demonstration. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9450–9458 (2019)
36. Wiles, O., Sophia Koepke, A., Zisserman, A.: X2face: A network for controlling face generation using images, audio, and pose codes. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 670–686 (2018)
37. Zadeh, A., Chong Lim, Y., Baltrusaitis, T., Morency, L.P.: Convolutional experts constrained local model for 3d facial landmark detection. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 2519–2528 (2017)
38. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9459–9468 (2019)
39. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)
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 CycleGAN

We train CycleGAN [39] to convert images from the synthetic domain (SynthFace [2]) to the real domain (FFHQ [15]). 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 [19] to generate the controllability metrics: $C_{\text{pred}}(I_+) - C_{\text{pred}}(I_−)$ 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 $C_{\text{pred}}(I_+) - C_{\text{pred}}(I_−)$ 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 10 strengthens 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 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.

6.3 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.
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.
Fig. 10: Change of hair colour from blond to black using ConfigNet (top) and synthetic data processed by CycleGAN (bottom). CycleGAN fails to preserve facial attributes 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, $C_{\text{pred}}(I_+) = 1$, $C_{\text{pred}}(I_-) = 0$ and MD should be 0. The mean difference $C_{\text{pred}}(I_+) - C_{\text{pred}}(I_-)$ gives the dynamic range of a given attribute, the higher it is the more controllable the attribute.

| Method          | $C_{\text{pred}}(I_+)$ | $C_{\text{pred}}(I_-)$ | MD     | $C_{\text{pred}}(I_+) - C_{\text{pred}}(I_-)$ |
|-----------------|-------------------------|-------------------------|--------|-----------------------------------------------|
| 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                                          |

described in [21]. In the image discriminators, we also use the style discriminator loss $L_{\text{style}}$ described in HoloGAN [23], while in the generator we add the identity loss $L_{\text{identity}}$ 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 $\lambda_{\text{eye}} = 5$, domain adverserial loss weight $\lambda_{\text{DA}} = 5$, identity loss weight $\lambda_{\text{identity}} = 10$, gradient penalty loss weight $\lambda_{\text{R1}} = 10$, perceptual loss weight in 1st stage $\lambda_{\text{perc}} = 0.0001$, perceptual loss weight in 2nd stage $\lambda_{\text{perc}} = 0.0005$. The adverserial losses on the images and style discriminator losses all have weight 1.

In the first training stage we sample $z \sim \mathcal{N}(0, I)$ and $r_R \sim \mathcal{U}(-r_{\text{lim}}, r_{\text{lim}})$, where $r_R$ is the rotation sample for real data and $r_{\text{lim}}$ is a pre-determined, per axis rotation limit. In all our experiments we set $r_{\text{lim}}$ 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 $r_{\text{lim}}$ by using a tanh activation and multiplying the output by $r_{\text{lim}}$.

Table 4 shows the architecture of the generator network $G$. In each AdaIN [12] input the latent vector $z$ is processed by a 2-layer MLP. The volume rotation
### 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        | $1 \times 1$   | LReLU      | $16 \times 16 \times 512$ | -             |
| conv2d\_2        | $4 \times 4$   | LReLU      | $16 \times 16 \times 256$ | AdaIN         |
| upsampling       | -              | -          | $32 \times 32 \times 256$ | -             |
| conv2d\_3        | $4 \times 4$   | LReLU      | $32 \times 32 \times 64$ | AdaIN         |
| upsampling       | -              | -          | $64 \times 64 \times 64$ | -             |
| conv2d\_4        | $4 \times 4$   | LReLU      | $64 \times 64 \times 32$ | AdaIN         |
| upsampling       | -              | -          | $128 \times 128 \times 32$ | -             |
| conv2d\_5        | $4 \times 4$   | LReLU      | $128 \times 128 \times 32$ | AdaIN         |
| upsampling       | -              | -          | $256 \times 256 \times 32$ | -             |
| conv2d\_6        | $4 \times 4$   | tanh       | $256 \times 256 \times 3$ | -             |

### Table 5: Architecture of the image discriminators $D_R, D_S$

| 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                    | -             |

The layer is the same as the one used in HoloGAN [23]. Table 5 shows the architecture of the image discriminators $D_R, D_S$. Following [23], most of the convolutional layers of the discriminator use instance normalization [34]. The latent GAN generator $G_{lat}$ and discriminator share the same 3-layer MLP architecture.

The networks are optimized using Adam [17] 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 [14], 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.
Table 6: Dimensionalities and descriptions of latent space factors

| Factor name     | dim $\theta_i$ | dim $z_i$ | Description of $\theta_i$ |
|-----------------|----------------|-----------|---------------------------|
| beard style     | 9              | 7         | PCA coefficients          |
| eyebrow style   | 44             | 7         | PCA coefficients          |
| expression      | 52             | 30        | 3D head model parameters  |
| 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   | 3              | 2         | one-hot encoding          |
| texture         | 50             | 30        | VAE latent space vector   |
| upper eyelash   | 3              | 2         | one-hot encoding          |

6.4 Factorized latent space details

Table 6 shows the dimensionalities of all latent space factors $z_i$ and corresponding synthetic data parameters $\theta_i$. The dimensionalities of each $z_i$ were chosen based 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 [2] and one additional dimension for the rotation of the jaw bone that leads to mouth opening.

6.5 Controllability metric details

The attribute predictor $C_{pred}$ we use for the metrics is a MobilenetV2 [29] 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 $\theta_i$ 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 $\theta_i$ with the chosen value.
6.6 Entanglement in synthetic dataset

While SynthFace allows for disentangling many face attributes, we have found that some of its 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 [12]. 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.

6.7 Additional figures

![Fig. 11: Modification of various attributes of faces from the 300-VW dataset using ConfigNet fine-tuned on the input image.](image-url)
Fig. 12: 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 $\theta_i$. Note the change of appearance when modifying head shape, discussion in Section 4.4.
Fig. 13: Results of sampling the latent space $z = E_R(I_R)$, where $I_R$ is randomly selected from FFHQ [15].

Fig. 14: Interpolation of two face attribute at a time in images sampled from the FFHQ [15] dataset. Note that the illumination direction stays constant as the head rotates.