

StyleRig: Rigging StyleGAN for 3D Control over Portrait Images

Ayush Tewari¹ Mohamed Elgharib¹ Gaurav Bharaj² Florian Bernard¹
Hans-Peter Seidel¹ Patrick Pérez³ Michael Zollhöfer⁴ Christian Theobalt¹

¹MPI Informatics, Saarland Informatics Campus ²Technicolor ³Valeo.ai ⁴Stanford University

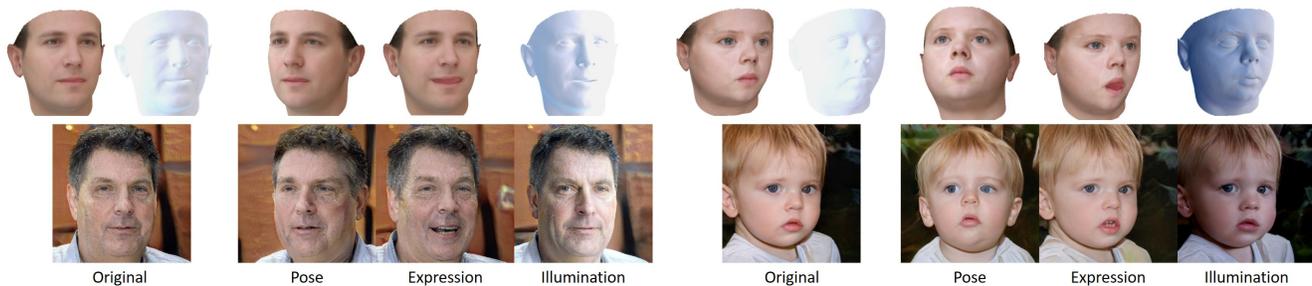


Figure 1: StyleRig allows for face rig-like control over StyleGAN generated portrait images, by translating semantic edits on 3D face meshes to the input space of StyleGAN.

Abstract

StyleGAN [18] generates photorealistic portrait images of faces with eyes, teeth, hair and context (neck, shoulders, background), but lacks a rig-like control over semantic face parameters that are interpretable in 3D, such as face pose, expressions, and scene illumination. Three-dimensional morphable face models (3DMMs) [10] on the other hand offer control over the semantic parameters, but lack photorealism when rendered and only model the face interior, not other parts of a portrait image (hair, mouth interior, background). We present the first method to provide a face rig-like control over a pretrained and fixed StyleGAN via a 3DMM. A new rigging network, RigNet is trained between the 3DMM’s semantic parameters and StyleGAN’s input. The network is trained in a self-supervised manner, without the need for manual annotations. At test time, our method generates portrait images with the photorealism of StyleGAN and provides explicit control over the 3D semantic parameters of the face.

1. Introduction

Photorealistic synthesis of portrait face images finds many applications in several fields including special effects, extended reality, virtual worlds, and next-generation communication. During the content creation process for such applications, artist control over the *face rig*’s semantic parameters, such as geometric identity, expressions, reflectance, or scene

illumination is desired. The computer vision and graphics communities have a rich history of modeling face rigs [21, 25, 26, 31]. These models provide artist-friendly control (often called a face rig), while navigating the various parameters of a morphable face model (3DMM) [3, 4]. Such methods are often limited by the lack of training data, and more importantly, lack of photorealism in the final rendering.

Through 3D face scanning techniques high-quality face geometry datasets can be obtained [5, 21]. However, models derived from these datasets are bound by the diversity of faces scanned and may limit the generalization over the rich set of human faces’ semantic parameterization. Further, deep learning-based models trained on *in-the-wild* data [31, 32, 35] also often rely on data-driven priors and other forms of regularization obtained from scan-based datasets. With respect to photorealism, perceptual losses recently showed an improvement of face modeling quality [9, 35] over existing methods. However, they still do not engender photorealistic face renders. Mouth interiors, hair, or eyes, let alone image background are often not modeled by such approaches. Generative Adversarial Networks (GANs) [14] have lately achieved photorealism [15, 17], especially for faces. Karras *et al.* [17] show that through a progressive growth of GAN’s generator and discriminator, one can better stabilize and speed up training. When trained on the CelebA-HQ [17] dataset this yields a remarkable level of photorealism for faces. Their approach also shows how photorealistic face images of non-existent people can be sampled from the learned GAN distribution. Building on Karras *et*

al. [17], StyleGAN [18] uses ideas from the style transfer literature [12, 29] and proposes an architecture capable of disentangling various face attributes. Promising results of control over various attributes, including coarse (hair, geometry), medium (expressions, facial hair) and fine (color distribution, freckles) attributes were shown. However, these controllable attributes are not semantically well defined, and contain several similar yet entangled semantic attributes. For example, both coarse and medium level attributes contain face identity information. In addition, the coarse levels contain several entangled attributes such as face identity and head pose.

We present a novel solution to *rig* StyleGAN using a semantic parameter space for faces. Our approach brings the best of both worlds: the controllable parametric nature of existing morphable face models [26, 31], and the high photorealism of generative face models [17, 18]. We employ a fixed and pretrained StyleGAN and do not require more data for training. Our focus is to provide computer graphics style rig-like control over the various semantic parameters. Our novel training procedure is based on a self-supervised two-way cycle consistency loss that is empowered by the combination of a face reconstruction network with a differentiable renderer. This allows us to measure the photometric re-rendering error in the image domain and leads to high quality results. We show compelling results of our method, including interactive control of StyleGAN generated imagery as well as image synthesis conditioned on well-defined semantic parameters.

2. Related Work

In the following, we discuss deep generative models for the synthesis of imagery with a focus on faces, as well as 3D parametric face models. For an in-depth overview of parametric face models and their possible applications we refer to the recent survey papers [10, 39].

Deep Generative Models Generative adversarial networks (GANs) contain two main blocks: a generator and a discriminator [14]. The generator takes a noise vector as an input and produces an output, and tries to fool the discriminator, whose purpose is to classify whether the output is real or fake. When the input to the network is a noise vector, the output is a sample from the learned distribution. Karras *et al.* [17] show that such a noise vector can generate high-resolution photorealistic images of human faces. To achieve this they employ a progressive strategy of slowly increasing the size of the generator and the discriminator, by adding more layers during training. This enables more stable training phase, and in turn helps learn high-resolution images of faces. StyleGAN [18] can synthesize highly photorealistic images while allowing for more control over the output, compared to Karras *et al.* [17]. However, StyleGAN

still suffers from a clear entanglement of semantically different attributes. Therefore, it does not provide a semantic and interpretable control over the image synthesis process. Exploring the latent space of GANs for image editing has been recently explored in Jahanian *et al.* [16]. They can only achieve simple transformations, such as zoom and 2D translations as they need ground truth images for each transformation during training. For faces, concurrent efforts have been made in controlling images synthesized by GANs [1, 30], but they lack explicit rig-like 3D control of the generative model. Isola *et al.* [15] use conditional GANs to produce image-to-image translations. Here, the input is not a noise vector, but a conditional image from a source domain, which is translated to the target domain by the generator. Their approach, however, requires paired training data. CycleGAN [38] and UNIT [22] learn to perform image-to-image translation only using unpaired data using cycle-consistency losses. GAUGAN [24] shows interactive semantic image synthesis based on spatially adaptive normalization. The remarkable quality achieved by GANs has inspired the development of several neural rendering applications for faces [10, 39] and others objects [8, 23, 36].

3D Morphable Models 3D Morphable Models (3DMMs) are commonly used to represent faces [3, 4]. Here, faces are parameterized by the identity geometry, expressions, skin reflectance and scene illumination. Expressions are commonly modeled using blendshapes, and illumination is generally modeled via spherical harmonics parameters [33]. The models are learned from 3D scans of people [5, 21], or more recently from in-the-wild internet footage [31]. The parametric nature of 3DMMs allows navigating and exploring the space of plausible faces, e.g., in terms of geometry, expressions and so on. Thus, synthetic images can be rendered based on different parameter configurations. The rendered images, however, often look synthetic and lack photorealism. More recently, neural rendering has been used to bridge the gap between synthetic computer graphics renderings and corresponding real versions [19, 34]. Several methods have been proposed for fitting face models to images [7, 11, 13, 20, 25, 26, 28, 31, 32, 33, 35]. Our work, however, focuses on learning-based approaches, that can be categorized into reconstruction only techniques [20, 25, 26, 28, 33], and reconstruction plus model learning [31, 32, 35]. MOFA [33] projects a face into the 3DMM space using a CNN, followed by a differentiable renderer to synthesize the reconstructed face. The network is trained in a self-supervised manner based on a large collection of face images. Tran *et al.* [35] use a perceptual loss to enhance the renderings of the reconstruction. RingNet [26] and FML [31] impose multi-image consistency losses to enforce identity similarity. RingNet also enforces identity dissimilarity between pictures of different people. Several approaches learn to reconstruct the

parameters of a 3DMM by training it on large scale synthetic data [20, 25, 28]. For a more comprehensive overview of all techniques please refer to [10, 39].

3. Overview

StyleGAN [18] can be seen as a function that maps a latent code $\mathbf{w} \in \mathbb{R}^l$ to a realistic portrait image $\mathbf{I}_{\mathbf{w}} = \text{StyleGAN}(\mathbf{w}) \in \mathbb{R}^{3 \times w \times h}$ of a human face. While the generated images are of very high quality and at a high resolution ($w = h = 1024$), there is no semantic control over the generated output, such as the head pose, expression, or illumination. StyleRig allows us to obtain a rig-like control over StyleGAN-generated facial imagery in terms of semantic and interpretable control parameters (Sec. 8). In the following, we explain the semantic control space (Sec. 4), training data (Sec. 5), network architecture (Sec. 6) and loss function (Sec. 7).

4. Semantic Rig Parameters

Our approach uses a parametric face model to achieve an explicit rig-like control of StyleGAN-generated imagery based on a set of semantic control parameters. The control parameters are a subset of $\mathbf{p} = (\alpha, \beta, \delta, \gamma, \mathbf{R}, \mathbf{t}) \in \mathbb{R}^f$, which describes the facial shape $\alpha \in \mathbb{R}^{80}$, skin reflectance $\beta \in \mathbb{R}^{80}$, facial expression $\delta \in \mathbb{R}^{64}$, scene illumination $\gamma \in \mathbb{R}^{27}$, head rotation $\mathbf{R} \in SO(3)$, and translation $\mathbf{t} \in \mathbb{R}^3$, with the dimensionality of \mathbf{p} being $f = 257$. We define the control space for the facial shape α and skin reflectance β using two low-dimensional affine models that have been computed via Principal Component Analysis (PCA) based on 200 (100 male, 100 female) scans of human faces [4]. The output of this model is represented by a triangle mesh with 53k vertices and per-vertex color information. The control space for the expression δ is given in terms of an additional affine model that captures the expression dependent displacement of the vertices. We obtain this model by applying PCA to a set of blendshapes [2, 6] which have been transferred to the topology of the shape and reflectance models. The affine models for shape, appearance, and expression cover more than 99% of the variance in the original datasets. Illumination γ is modeled based on three bands of spherical harmonics per color channel leading to an additional 27 parameters.

5. Training Corpus

Besides the parametric face model, our approach requires a set of face images $\mathbf{I}_{\mathbf{w}}$ and their corresponding latent codes \mathbf{w} as training data. To this end, we sample $N = 200k$ latent codes $\mathbf{w} \in \mathbb{R}^l$ and generate the corresponding photorealistic face images $\mathbf{I}_{\mathbf{w}} = \text{StyleGAN}(\mathbf{w})$ using a pretrained StyleGAN network. We use $l = 18 \times 512$ dimensional latent

space which is the output of the mapping network in StyleGAN, as it has been shown to be more disentangled [1, 18]. Here, 18 latent vectors of size 512 are used at different resolutions. Each training sample is generated by combining up to 5 separately sampled latent vectors, similar to the mixing regularizer in Karras *et al.* [18]. This allows our networks to reason independently about the latent vectors at different resolutions. Given these $(\mathbf{w}, \mathbf{I}_{\mathbf{w}})$ pairs, our approach can be trained in a self-supervised manner without requiring any additional image data or manual annotations.

6. Network Architecture

Given a latent code $\mathbf{w} \in \mathbb{R}^l$ that corresponds to an image $\mathbf{I}_{\mathbf{w}}$, and a vector $\mathbf{p} \in \mathbb{R}^f$ of semantic control parameters, we want to learn a function that outputs a modified latent code $\hat{\mathbf{w}} = \text{RigNet}(\mathbf{w}, \mathbf{p})$. The modified latent code $\hat{\mathbf{w}}$ should map to a modified face image $\mathbf{I}_{\hat{\mathbf{w}}} = \text{StyleGAN}(\hat{\mathbf{w}})$ that obeys the control parameters \mathbf{p} . One example would be changing the rotation of the face in an image such that it matches a given target rotation, while maintaining the facial identity, expression, and scene illumination (see Sec. 8 for examples). We train separate RigNet networks for the different modes of control i.e., pose, expressions and illumination. RigNet is implemented based on a linear two-layer perceptron (MLP). We propose a self-supervised training of RigNet based on two-way cycle consistency losses and a differentiable face reconstruction (DFR) network. Fig. 2 shows an overview of our architecture. Our network combines several components that fulfill specific tasks.

Differentiable Face Reconstruction One key component is a pretrained differentiable face reconstruction (DFR) network. This parameter regressor is a function $\mathcal{F} : \mathbb{R}^l \rightarrow \mathbb{R}^f$ that maps a latent code \mathbf{w} to a vector of semantic control parameters $\mathbf{p}_{\mathbf{w}} = \mathcal{F}(\mathbf{w})$. In practice, we model \mathcal{F} using a three layer MLP with ELU activations after every intermediate layer, and train it in a self-supervised manner. This requires a differentiable render layer $\mathcal{R} : \mathbb{R}^f \rightarrow \mathbb{R}^{3 \times w \times h}$ that takes a face parameter vector \mathbf{p} as input, converts it into a 3D mesh and generates a synthetic rendering $\mathbf{S}_{\mathbf{w}} = \mathcal{R}(\mathbf{p}_{\mathbf{w}})$ of the face¹. We then train \mathcal{F} using a rerendering loss:

$$\mathcal{L}_{\text{render}}(\mathbf{I}_{\mathbf{w}}, \mathbf{p}) = \mathcal{L}_{\text{photo}}(\mathbf{I}_{\mathbf{w}}, \mathbf{p}) + \lambda_{\text{land}} \mathcal{L}_{\text{land}}(\mathbf{I}_{\mathbf{w}}, \mathbf{p}) . \quad (1)$$

The first term is a dense photometric alignment loss:

$$\mathcal{L}_{\text{photo}}(\mathbf{I}_{\mathbf{w}}, \mathbf{p}) = \|\mathbf{M} \odot (\mathbf{I}_{\mathbf{w}} - \mathcal{R}(\mathbf{p}))\|_2^2 .$$

Here, \mathbf{M} is a binary mask with all pixels where the face mesh is rendered set to 1 and \odot is element-wise multiplication. We also use a sparse landmark loss

$$\mathcal{L}_{\text{land}}(\mathbf{I}_{\mathbf{w}}, \mathbf{p}) = \|\mathbf{L}_{\mathbf{I}_{\mathbf{w}}} - \mathbf{L}_{\mathbf{M}}\|_2^2 ,$$

¹We use point-based rendering of the mesh vertices.

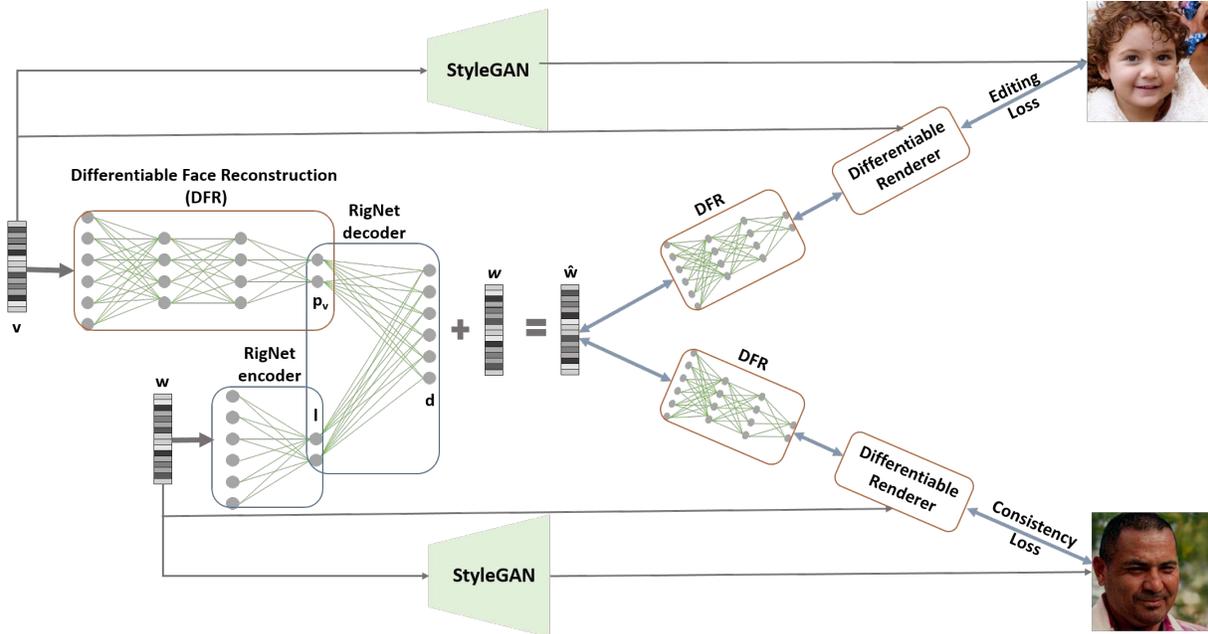


Figure 2: StyleRig enables rig-like control over StyleGAN-generated facial imagery based on a learned rigger network (RigNet). To this end, we employ a self-supervised training approach based on a differentiable face reconstruction (DFR) and a neural face renderer (StyleGAN). The DFR and StyleGAN networks are pretrained and their weights are fixed, only RigNet is trainable. We define the consistency and edit losses in the image domain using a differentiable renderer.



Figure 3: Differentiable Face Reconstruction. Visualized are (image, reconstruction) pairs. The network however, only gets the latent vector corresponding to the images as input.

where $\mathbf{L}_{I_w} \in \mathbb{R}^{66 \times 2}$ are 66 automatically computed landmarks [27] on the image I_w , and \mathbf{L}_M are the corresponding landmark positions on the rendered reconstructed face. The landmark vertices on the mesh are manually annotated before training. λ_{land} is a fixed weight used to balance the loss terms. In addition, we also employ statistical regularization on the parameters of the face model, as done in MoFA. [33]. After training, the weights of \mathcal{F} are fixed. Fig. 3 shows some results of the reconstructions obtained by DFR.

RigNet Encoder The encoder takes the latent vector \mathbf{w} as input and linearly transforms it into a lower dimensional vector \mathbf{l} of size 18×32 . Each sub-vector \mathbf{w}_i of \mathbf{w} of size 512 is independently transformed into a sub-vector \mathbf{l}_i of size 32, for all $i \in \{0, \dots, 17\}$.

RigNet Decoder The decoder transforms \mathbf{l} and the input control parameters \mathbf{p} into the output $\hat{\mathbf{w}}$. Similar to the encoder, we use independent linear decoders for each \mathbf{l}_i . Each layer first concatenates \mathbf{l}_i and \mathbf{p} , and transforms it into \mathbf{d}_i ,

for all $i \in \{0, \dots, 17\}$. The final output is computed as $\hat{\mathbf{w}} = \mathbf{d} + \mathbf{w}$.

7. Self-supervised Training

Our goal is to train RigNet such that we can inject a subset of parameters into a given latent code \mathbf{w} . For example, we might want to inject a new head pose, while maintaining the facial identity, expression, and illumination in the original image synthesized from \mathbf{w} . We employ the following loss function for training:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{edit}} + \mathcal{L}_{\text{consist}} . \quad (2)$$

It consists of a reconstruction loss \mathcal{L}_{rec} , an editing loss $\mathcal{L}_{\text{edit}}$, and a consistency loss $\mathcal{L}_{\text{consist}}$. Since we do not have ground truth for the desired modifications (our training corpus only contains one image per person), we employ self-supervision based on cycle-consistent editing and consistency losses. We optimize $\mathcal{L}_{\text{total}}$ based on AdaDelta [37] with a learning rate of 0.01. In the following, we provide details.

Reconstruction Loss We want to design RigNet such that it reproduces the latent codes in the training corpus. Formally, we want that $\text{RigNet}(\mathbf{w}, \mathcal{F}(\mathbf{w})) = \mathbf{w}$. We enforce this with the following ℓ_2 -loss:

$$\mathcal{L}_{\text{rec}} = \|\text{RigNet}(\mathbf{w}, \mathcal{F}(\mathbf{w})) - \mathbf{w}\|_2^2 .$$

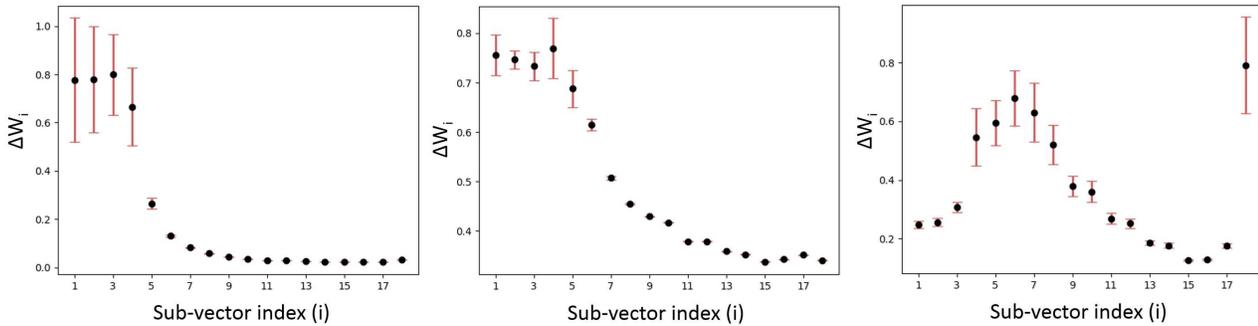


Figure 4: Change of latent vectors at different resolutions. Coarse vectors are responsible for rotation (left), medium for expressions (middle), medium and fine for illumination (right).

This constraint anchors the learned mapping at the right location in the latent space. Without this constraint, learning the mapping is underconstrained, which leads to a degradation in the image quality (see Sec. 8). Since \mathcal{F} is pretrained and not updated, the semantics of the control space are enforced.

Cycle-Consistent Per-Pixel Editing Loss Given two latent codes, \mathbf{w} and \mathbf{v} with corresponding images \mathbf{I}_w and \mathbf{I}_v , we transfer the semantic parameters of \mathbf{v} to \mathbf{w} during training. We first extract the target parameter vector $\mathbf{p}_v = \mathcal{F}(\mathbf{v})$ using the differentiable face reconstruction network. Next, we inject a subset of the parameters of \mathbf{p}_v (the ones we want to modify) into the latent code \mathbf{w} to yield a new latent code $\hat{\mathbf{w}} = \text{RigNet}(\mathbf{w}, \mathbf{p}_v)$, so that $\mathbf{I}_{\hat{\mathbf{w}}} = \text{StyleGAN}(\hat{\mathbf{w}})$ (ideally) corresponds to the image \mathbf{I}_w , modified according to the subset of the parameters of \mathbf{p}_v . For example, $\hat{\mathbf{w}}$ might retain the facial identity, expression and scene illumination of \mathbf{w} , but should perform the head rotation specified in \mathbf{p}_v .

Since we do not have ground truth for such a modification, i.e., the image $\mathbf{I}_{\hat{\mathbf{w}}}$ is unknown, we employ supervision based on a cycle-consistent editing loss. The editing loss enforces that the latent code $\hat{\mathbf{w}}$ contains the modified parameters. We enforce this by mapping from the latent to the parameter space $\hat{\mathbf{p}} = \mathcal{F}(\hat{\mathbf{w}})$. The regressed parameters $\hat{\mathbf{p}}$ should have the same rotation as \mathbf{p}_v . We could measure this directly in the parameter space but this has been shown to not be very effective [33]. We also observed in our experiments that minimizing a loss in the parameter space does not lead to desired results, since the perceptual effect of different parameters in the image space can be very different.

Instead, we employ a re-rendering loss similar to the one used for differentiable face reconstruction. We take the original target parameter vector \mathbf{p}_v and replace its rotation parameters with the regressed rotation from $\hat{\mathbf{p}}$, resulting in \mathbf{p}_{edit} . We can now compare this to \mathbf{I}_v using the re-rendering loss (see Eq. 1):

$$\mathcal{L}_{\text{edit}} = \mathcal{L}_{\text{render}}(\mathbf{I}_v, \mathbf{p}_{\text{edit}}) .$$

We do not use any regularization terms here. Such a loss function ensures that the rotation component of \mathbf{p}_{edit} aligns with \mathbf{I}_v , which is the desired output. The component of \mathbf{p}_v which is replaced from $\hat{\mathbf{p}}$ depends on the property we want to change. It could either be the pose, expressions, or illumination parameters.

Cycle-consistent Per-pixel Consistency Loss In addition to the editing loss, we enforce consistency of the parameters that should not be changed by the performed edit operation. The regressed parameters $\hat{\mathbf{p}}$ should have the same unmodified parameters as \mathbf{p}_w . Similarly as above, we impose this in terms of a re-rendering loss. We take the original parameter vector \mathbf{p}_w and replace all parameters that should not be modified by the regressed ones from $\hat{\mathbf{p}}$, resulting in $\mathbf{p}_{\text{consist}}$. In the case of modifying rotation values, the parameters that should not change are expression, illumination as well as identity parameters (shape and skin reflectance). This leads to the loss function:

$$\mathcal{L}_{\text{consist}} = \mathcal{L}_{\text{render}}(\mathbf{I}_w, \mathbf{p}_{\text{consist}}) .$$

Siamese Training Since we have already sampled two latent codes \mathbf{w} and \mathbf{v} during training, we perform the same operations in a reverse order, i.e., in addition to injecting \mathbf{p}_v into \mathbf{w} , we also inject \mathbf{p}_w into \mathbf{v} . To this end, we use a Siamese network with two towers that have shared weights. This results in a two-way cycle consistency loss.

8. Results

At test time, StyleRig allows control over the pose, expression, and illumination parameters of StyleGAN generated images. We demonstrate the efficacy of our approach with three applications: Style Mixing (8.1), Interactive Rig Control (8.2) and Conditional Image Generation (8.3).

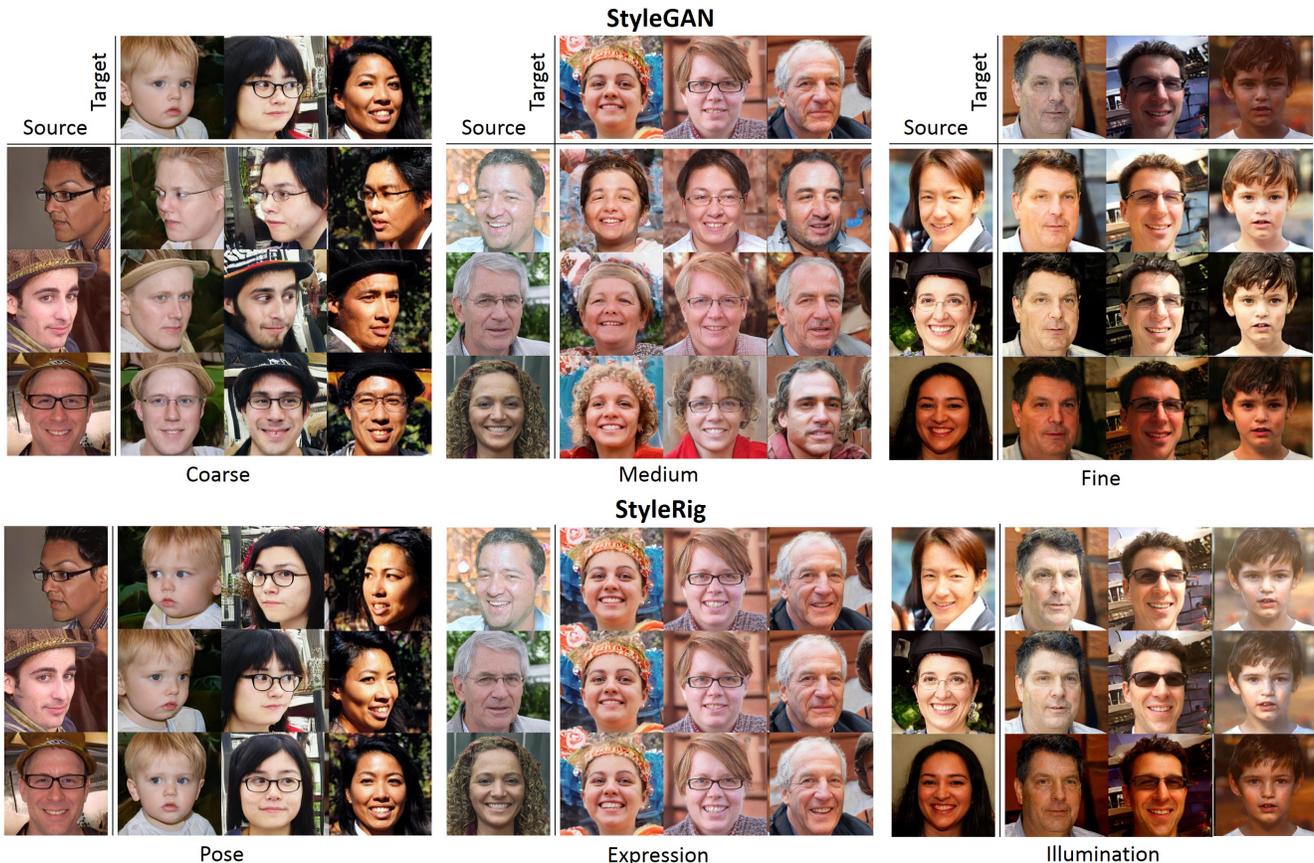


Figure 5: Mixing between source and target images generated by StyleGAN. For StyleGAN, the latent vectors of the source samples (rows) are copied to the target vectors (columns). StyleRig allows us to mix semantically meaningful parameters, i.e., head pose, expressions and scene illumination. These parameters can be copied over from the source to target images.

8.1. Style Mixing

Karras *et al.* [18] show StyleGAN vectors at different scales that correspond to different *styles*. To demonstrate *style mixing*, latent vectors at certain resolutions are copied from a source to a target image, and new images are generated. As shown in Fig. 5, coarse styles contain information about the pose as well as identity, medium styles include information about expressions, hair structure, and illumination, while fine styles include the color scheme of the source. We show a similar application of mixing, but with significantly more complete control over the semantic parameters. To generate images with a target identity, we transfer the source parameters of our face rig to the target latent, resulting in images with different head poses, expressions and illumination. This rig-like control is not possible via the mixing strategy of Karras *et al.* which entangles multiple semantic dimensions in the mixed results. In Fig. 4, we analyze how the latent vectors of StyleGAN are transformed by StyleRig. The figure shows the average change and variance (change is measured as ℓ_2 distance) of StyleGAN latent vectors at

all resolutions, computed over 2500 mixing results. As expected, coarse latent code vectors are mainly responsible for rotation. Expression is controlled both by coarse and medium level latent codes. The light direction is mostly controlled by the medium resolution vectors. However, the fine latent vector also plays an important role in the control of the global color scheme of the images. Rather than having to specify which vectors need to change and by how much, StyleRig recovers this mapping in a self-supervised manner. As shown in Fig. 5, we can also preserve scene context like background, hair styles and accessories better.

8.2. Interactive Rig Control

Since the parameters of the 3DMM can also be controlled independently, StyleRig allows for explicit semantic control of StyleGAN generated images. We develop a user interface where a user can interact with a face mesh by interactively changing its pose, expression, and scene illumination parameters. These updated parameters are then fed into RigNet to generate new images at interactive frame rates (~ 5 fps).

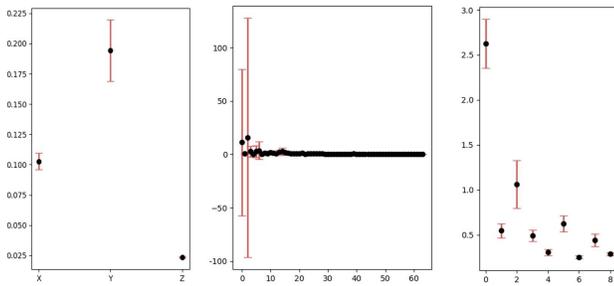


Figure 6: Distribution of face model parameters in the training data. x -axis shows the face model parameters for rotation, expression and illumination from left-right. y -axis shows the mean and variance of the parameters computed over $20k$ training samples.

Fig. 1 shows the results for various controls over StyleGAN images: pose, expression, and illumination edits. The control rig carries out the edits in a smooth interactive manner. Please refer to the supplemental video for more results.

Analysis of StyleRig The interactive editor allows us to easily *inspect* the trained networks. We observe that while the network does a good job at most controls, some expressivity of the 3D parametric face model is lost. That is, RigNet cannot transfer all modes of parametric control to similar changes in the StyleGAN generated images. For example, we notice that in-plane rotation of the face mesh is ignored. Similarly, many expressions of the face mesh do not translate well into the resultant generated images. We attribute these problems to the bias in the images StyleGAN has been trained on. To analyze these modes, we look at the distribution of face model parameters in our training data, generated from StyleGAN, see Fig. 6. We notice that in-plane rotations (rotation around the Z-axis) are hardly present in the data. In fact, most variation is only around the Y-axis. This could be because StyleGAN is trained on the Flickr-HQ dataset [18]. Most static images of faces in such a dataset would not include in-plane rotations. The same reasoning can be applied to expressions, where most generated images consist of either neutral or smiling/laughing faces. These expressions can be captured using up to three blendshapes. Even though the face rig contains 64 vectors, we cannot control them well because of the biases in the distribution of the training data. Similarly, the lighting conditions are also limited in the dataset. We note that there are larger variations in the global color and azimuth dimensions, as compared to the other dimensions. Our approach provides an intuitive and interactive user interface which allows us to inspect not only StyleRig, but also the biases present in StyleGAN.



Figure 7: Explicit control over the 3D parameters allows us to turn StyleGAN into a conditional generative model.

8.3. Conditional Image Generation

Explicit and implicit control of a pretrained generative model allows us to turn it into a conditional one. We can simply fix the pose, expression, or illumination inputs to RigNet in order to generate images which correspond to the specified parameters, see Fig. 7. This is a straight forward way to convert an unconditional generative model into a conditional model, and can produce high-resolution photo-realistic results. It is also very efficient, as it takes us less than 24 hours to train StyleRig, while training a conditional generative model from scratch should take at least as much time as StyleGAN, which takes more than 41 days to train (both numbers are for an Nvidia Volta GPU).

8.4. Comparisons to Baseline Approaches

In the following, we compare our approach with several baseline approaches.

“Steering” the latent vector Inspired by Jahanian *et al.* [16], we design a network architecture which tries to *steer* the StyleGAN latent vector based on the change in parameters. This network architecture does not use the latent vector w as an input, and thus does not require an encoder. The inputs to the network are the delta in the face model parameters, with the output being the delta in the latent vector. In our settings, such an architecture does not lead to desirable results with the network not being able to deform the geometry of the faces, see Fig. 8. Thus, the semantic deltas in latent space should also be conditional on the the latent vectors, in addition to the target parameters.

Different Loss Functions As explained in Eq. 2, our loss function consists of three terms. For the first baseline, we switch off the reconstruction loss. This can lead to the output latent vectors drifting from the space of StyleGAN latent codes, thus resulting in non-face images. Next, we switch off the consistency loss. This loss term enforces the consistency of all face model parameters, other than the one being changed. Without this term, changing one dimension, for example the illumination, also changes others such as the head pose. Our final model ensures the desired edits

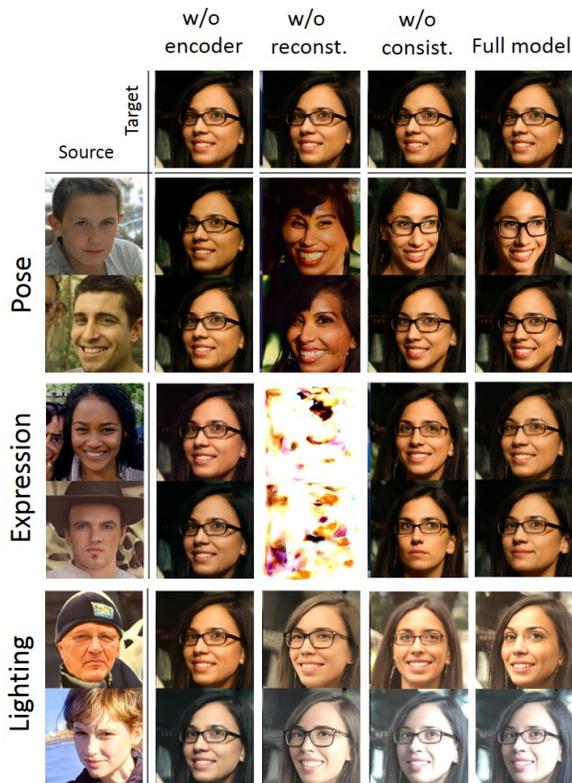


Figure 8: Baseline comparisons. Our full approach obtains the highest quality results.

with consistent identity and scene information. Note that switching off the editing loss is not a good baseline, as it would not add any control over the generator.

8.5. Simultaneous Parameter Control

In addition to controlling different parameters independently, we can also control them simultaneously. To this end, we train RigNet, such that, it receives target pose, expression, and illumination parameters as input. For every (\mathbf{w}, \mathbf{v}) training code vector pair, we sample three training samples. Here, one out of the three parameters (pose, expression or illumination) is changed in each sample. We then use the loss function defined in Eq. 2 for each such sample. Thus, RigNet learns to edit each dimension of the control space independently, while also being able to combine the edits using the same network. Fig. 9 shows mixing results where pose, expression and illumination parameters are transferred from the source to target images.

9. Limitations

While we have demonstrated high quality semantic control of StyleGAN-generated facial imagery, our approach is still subject to a few limitations that can be addressed in follow-up work. In the analysis sections, we have already



Figure 9: RigNet can also control pose, expression, and illumination parameters simultaneously. These parameters are transferred from source to target images, while the identity in the target images is preserved.

discussed that StyleRig is not able to exploit the full expressivity of the parametric face model. This provides a nice insight into the inner workings of StyleGAN and allows us to introspect the biases it learned. In the future, this might lead the ways to designing better generative models. Our approach is also limited by the quality of the employed differentiable face reconstruction network. Currently, this model does not allow us to reconstruct fine-scale detail, thus we can not explicitly control them. Finally, there is no explicit constraint that tries to preserve parts of the scene that are not explained by the parametric face model, e.g., the background or hair style. Therefore, these parts can not be controlled and might change when editing the parameters.

10. Conclusion

We have proposed StyleRig, a novel approach that provides face rig-like control over a pretrained and fixed StyleGAN network. Our network is trained in a self-supervised manner and does not require any additional images or manual annotations. At test time, our method generates images of faces with the photorealism of StyleGAN, while providing explicit control over a set of semantic control parameters. We believe that the combination of computer graphics control with deep generative models enables many exciting editing applications, provides insights into the inner workings of the generative model, and will inspire follow-up work.

Acknowledgements: We thank True-VisionSolutions Pty Ltd for providing the 2D face tracker. This work was supported by the ERC Consolidator Grant 4DReply (770784), the Max Planck Center for Visual Computing and Communications (MPC-VCC), and by Technicolor.

References

- [1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to embed images into the stylegan latent space? In *International Conference on Computer Vision (ICCV)*, 2019.
- [2] Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. The digital emily project: Photoreal facial modeling and animation. In *ACM SIGGRAPH Courses*, pages 12:1–12:15, 2009.
- [3] Volker Blanz, Curzio Basso, Tomaso Poggio, and Thomas Vetter. Reanimating faces in images and video. In *Computer graphics forum*, pages 641–650. Wiley Online Library, 2003.
- [4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH's Computer Graphics and Interactive Techniques*, pages 187–194, 1999.
- [5] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3D morphable models. *International Journal of Computer Vision (IJCV)*, 126(2):233–254, Apr. 2018.
- [6] Chen Cao, Yanlin Weng, Shun Zhou, Yiyong Tong, and Kun Zhou. Facewarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 20(3):413–425, Mar. 2014.
- [7] Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 35(4):126:1–126:12, 2016.
- [8] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *International Conference on Computer Vision (ICCV)*, 2019.
- [9] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set. In *CVPR Workshops*, 2019.
- [10] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models – past, present and future, 2019.
- [11] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Reconstruction of personalized 3D face rigs from monocular video. *ACM Trans. on Graph. (Proceedings of SIGGRAPH)*, 35(3):28:1–15, June 2016.
- [12] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.
- [13] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018.
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680. 2014.
- [15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.
- [16] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.
- [17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representation (ICLR)*, 2018.
- [18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CVPR*, 2019.
- [19] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, N. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. Deep Video Portraits. *ACM Trans. on Graph. (Proceedings of SIGGRAPH)*, 2018.
- [20] Hyeonwoo Kim, Michael Zollhöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Christian Theobalt. Inverse-FaceNet: Deep Single-Shot Inverse Face Rendering From A Single Image. In *CVPR*, 2018.
- [21] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. FLAME: Learning a model of facial shape and expression from 4d scans. *ACM Trans. on Graph.*, 36(6):194:1–194:17, 2017.
- [22] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks.
- [23] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello. Lookngood: Enhancing performance capture with real-time neural re-rendering. *ACM Trans. on Graph. (Proceedings of SIGGRAPH-Asia)*, 37(6):255:1–255:14, 2018.
- [24] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.
- [25] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *CVPR*, 2017.
- [26] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *CVPR*, pages 7763–7772, 2019.
- [27] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Deformable model fitting by regularized landmark mean-shift. *91(2):200–215*, 2011.
- [28] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation. In *ICCV*, 2017.
- [29] Ahmed Selim, Mohamed Elgharib, and Linda Doyle. Painting style transfer for head portraits using convolutional neural networks. pages 129:1–129:18, 2016.

- [30] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing, 2019.
- [31] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zollhoefer, and Christian Theobalt. FML: Face model learning from videos. 2019.
- [32] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeonwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *CVPR*, 2018.
- [33] Ayush Tewari, Michael Zollhöfer, Hyeonwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *ICCV*, pages 3735–3744, 2017.
- [34] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.
- [35] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphable model. In *CVPR*, June 2019.
- [36] Ye Yu and William A. P. Smith. InverseRenderNet: Learning single image inverse rendering. In *CVPR*, 2019.
- [37] Matthew D. Zeiler. Adadelta: An adaptive learning rate method, 2012.
- [38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.
- [39] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. *Comput. Graph. Forum (Eurographics State of the Art Reports 2018)*, 37(2), 2018.