Mensah, Dann; Kim, Nam Hee; Aittala, Miika; Laine, Samuli; Lehtinen, Jaakko

A Hybrid Generator Architecture for Controllable Face Synthesis

Published in:
Proceedings - SIGGRAPH 2023 Conference Papers

DOI:
10.1145/3588432.3591563

Published: 23/07/2023

Document Version
Publisher's PDF, also known as Version of record

A Hybrid Generator Architecture for Controllable Face Synthesis

Dann Mensah
dann.mensah@aalto.fi
Aalto University
Helsinki, Finland

Nam Hee Kim
namhee.kim@aalto.fi
Aalto University
Helsinki, Finland

Miika Aittala
maittala@nvidia.com
NVIDIA
Helsinki, Finland

Samuli Laine
slaine@nvidia.com
NVIDIA
Helsinki, Finland

Jaakko Lehtinen
jaakko.lehtinen@aalto.fi
Aalto University and
NVIDIA
Helsinki, Finland

ABSTRACT

Modern data-driven image generation models often surpass traditional graphics techniques in quality. However, while traditional modeling and animation tools allow precise control over the image generation process in terms of interpretable quantities — e.g., shapes and reflectances — endowing learned models with such controls is generally difficult.

In the context of human faces, we seek a data-driven generator architecture that simultaneously retains the photorealistic quality of modern generative adversarial networks (GAN) and allows explicit, disentangled controls over head shapes, expressions, identity, background, and illumination. While our high-level goal is shared by a large body of previous work, we approach the problem with a different philosophy: We treat the problem as an unconditional synthesis task, and engineer interpretable inductive biases into the model that make it easy for the desired behavior to emerge. Concretely, our generator is a combination of learned neural networks and fixed-function blocks, such as a 3D morphable head model and texture-mapping rasterizer, and we leave it up to the training process to figure out how they should be used together. This greatly simplifies the training problem by removing the need for labeled training data; we learn the distributions of the independent variables that drive the model instead of requiring that their values are known for each training image. Furthermore, we need no contrastive or imitation learning for correct behavior.

We show that our design successfully encourages the generative model to make use of the internal, interpretable representations in a semantically meaningful manner. This allows sampling of different aspects of the image independently, as well as precise control of the results by manipulating the internal state of the interpretable blocks within the generator. This enables, for instance, facial animation using traditional animation tools.

CCS CONCEPTS

• **Computing methodologies** → **Shape modeling**; **Rendering**; **Neural networks**; **Unsupervised learning**.

KEYWORDS

face modeling, generative adversarial networks, differentiable rendering

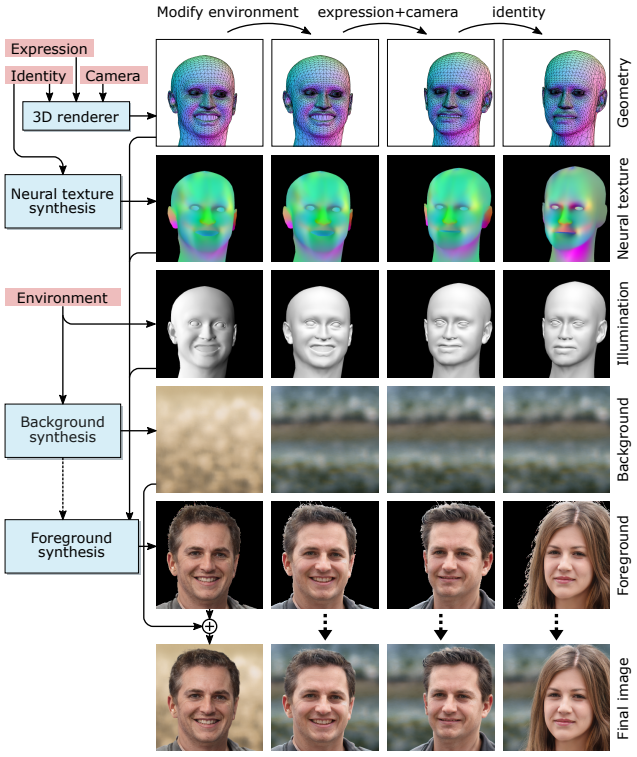


Figure 1: High-level view of our image synthesis process. Four separate latent vectors (red) influence different phases of the synthesis pipeline. A parametric head model is rendered to control the overall geometry based on identity, expression, and camera latents, and a neural texture (shown mapped onto the rendered head) encodes the spatial variation based on identity. An environment latent controls the illumination and background synthesis. A foreground synthesis network converts the intermediate representation into a photorealistic RGBA image, and the final image is composited using standard alpha blending. The result is photorealistic and can be controlled by manipulating the individual latents or the mesh directly.

SIGGRAPH '23 Conference Proceedings, August 06–10, 2023, Los Angeles, CA, USA

ACM ISBN 979-8-4007-0159-7/23/08.
https://doi.org/10.1145/3588432.3591563

ACM Reference Format:
Dann Mensah, Nam Hee Kim, Miika Aittala, Samuli Laine, and Jaakko Lehtinen. 2023. A Hybrid Generator Architecture for Controllable Face Synthesis. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings (SIGGRAPH '23 Conference Proceedings), August 06–10, 2023, Los Angeles, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3588432.3591563

1 INTRODUCTION

Data-driven generative models have quickly turned from curiosities to practical tools for image generation and editing. Current models are often able to synthesize images far more realistic than even the best traditional computer graphics techniques, with perhaps the only exception found in ultra-high-end film production where immense effort is expended to capture and model shapes, appearances, and movements used as inputs to physically based renderers.

Most data-driven generative image models are built on convolutional neural networks (CNN). We find their performance as surprising as it is astonishing — all these models do is hierarchically process local pixel neighborhoods, but they nonetheless learn to represent photorealistic three-dimensional objects in various poses, viewpoints, and illuminations. In particular, this is achieved without any specialized computational primitives tailored to perspective imaging or light-surface interaction.

In practice, the uses of data-driven generative models are hampered by their opaque nature. While the models often learn to disentangle effects such as pose and illumination to some degree, fine-grained control remains a challenging and active research topic.

Working in the context of human faces, we seek a learned generative model that retains the photorealistic fidelity obtainable from modern generative adversarial networks (GAN) while allowing control of the results by standard graphics approaches, such as rotating the view or changing the pose. Our focus is strictly limited to face synthesis — not editing. Our approach differs in two crucial ways from the many previous methods that aim at the same overall goal. First, we seek an architecture with inductive biases that encourage interpretable behavior to emerge as a byproduct of solving an unconditional modeling task. Second, instead of forcing interpretable controls over things such as camera poses, head shape parameters, etc., onto the generator by conditioning, we only learn their statistical distribution from data. Our approach has the significant benefit that we do not need to estimate conditioning parameters using inference models.

Specifically, we architect our generator (Figure 2) as a combination of CNNs, multi-layer perceptrons (MLPs), and the standard graphics operations of perspective projection, rasterization, texture mapping, and a 3D morphable face model (3DMM). Importantly, we do not seek a model that would directly output 3D assets that would yield photorealistic results when rendered using a traditional renderer. Instead, we embed interpretable computer graphics primitives deeper into the model and rely on the power of CNNs for generating the final pixels. All test-time control of the results is performed by modifying the interpretable internal representations inside the network, which we show to carry over to the final image.

In summary, we contribute a GAN architecture that learns to synthesize photorealistic faces with disentangled controls for identity, pose, and expression, trained with unlabeled data. The 3D morphable model and a perspective camera embedded deep inside the generative model enable control of many aspects of the results using standard graphics techniques, such as vertex animation.

2 RELATED WORK

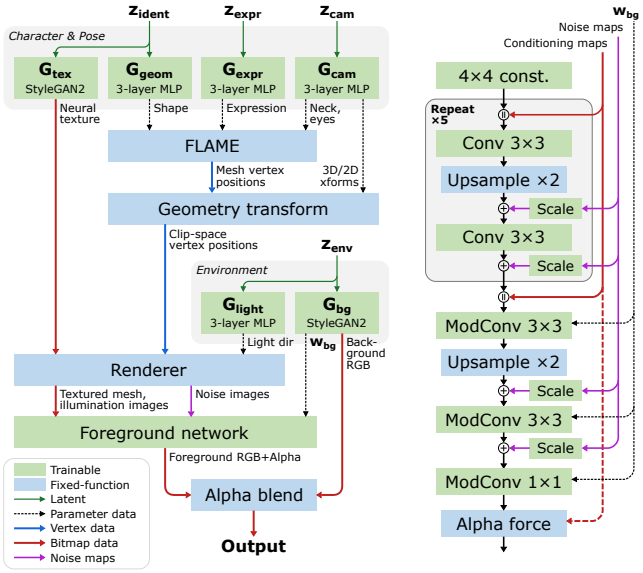
2.0.1 Controllable and Interpretable Face GANs. Recent years have seen an unprecedented advancement in GAN-based [Goodfellow et al. 2014] face synthesis thanks to the availability of data (e.g., [Karras et al. 2019; Liu et al. 2015]) and active research efforts (e.g. [Brock et al. 2018; Karras et al. 2021, 2019, 2020b; Sauer et al. 2022]). The impressive output image quality afforded by GANs led to rapidly growing interests in harnessing them for practical image generation and editing tools. In early efforts, variants of conditional GANs [Mirza and Osindero 2014] demonstrated their use case in identity-preserving generation in face synthesis (e.g., [Bao et al. 2017, 2018; Shen et al. 2018; Tran et al. 2017; Yin et al. 2017]). However, these formulations do not offer direct controls for camera pose, lighting, background properties, and face shape.

More recent efforts have focused on investigating the highly entangled latent space of GANs. InfoGAN [Chen et al. 2016] develops interpretable latent spaces without supervision, and various works restructure the latent space to enable interpretable control (e.g., [Härkönen et al. 2020; Kim et al. 2021]).

2.0.2 Training 3D Face Generators on 2D Images. Recent approaches have highlighted that imposing a layer structure, e.g., an alpha map, in the generator output improves its 3D understanding, even when the training signal is only in 2D (e.g. [Chen et al. 2019; Yang et al. 2022; Zhao et al. 2022]). To achieve control over practical variables such as pose and lighting, many recent efforts have additionally focused on utilizing 3D representations in training generative models. For example, HoloGAN [Nguyen-Phuoc et al. 2019] disentangles 3D pose from other content and supports view manipulation at inference time by requiring that 2D projections of a randomly rotated, abstract 3D feature grid results in realistic images. HoloGAN's architecture places only weak constraints on the 3D representation and does not offer interpretable conrols apart from rotation.

More recent work incorporates explicitly three-dimensional representations and blend analytic and neural rendering [Gecer et al. 2018; Thies et al. 2019]. They differ from our goals and methods in key aspects. EG3D [Chan et al. 2022] uses a 3D representation that is not tailored towards interpretability and fine-grained control; instead of a structured mesh, it constructs a radiance field based on a tri-plane representation. This offers the flexibility to model diverse geometries but offers no specific constraints or controls pertaining to face shapes, identity, pose, or lighting. GET3D [Gao et al. 2022] is conditioned on images and outputs textured meshes that can be rendered as-is using a traditional graphics pipeline. This ability comes at the price of significantly lower photorealism. StyleRig [Tewari et al. 2020] learns an additional rigging network, enabling semantic 3D controls for a pre-trained StyleGAN network. In our work, we strike a different balance by incorporating the graphics primitives deeper into the generator and letting learned components produce the final colors, effectively sidestepping the difficult attempt to produce pixel-perfect results by a traditional renderer.

In an attempt to represent faces more explicitly, DiscoFaceGAN [Deng et al. 2020] makes use of morphable 3D face models [Blanz and Vetter 1999; Paysan et al. 2009] in the training process and achieves inference-time manipulation of identity, deformation, lighting, and pose of the face in image. Despite the similar goals,



(a) Generator architecture (b) Foreground network

Figure 2: System diagram. (a) High-level architecture of our generator. The generator is composed of two StyleGAN2 networks, four simple MLP networks, and a custom foreground synthesis network, along with various fixed-function components. (b) Internal design of the foreground network. Symbols ⊕ and ⓘ denote summation and concatenation, respectively. See text for details.

the approach differs significantly from ours. First, the generator is a pure CNN that is conditioned on the control parameters of interest. Second, the training data needs to be labeled with the conditioning variables. These variables are extracted from the dataset using pre-trained face parsing networks [Lin et al. 2019], and their statistical distributions are modeled separately using pre-trained VAEs [Kingma and Welling 2014]. Disentanglement is then achieved by imposing explicit constraints in the form of a contrastive learning objective where synthetic renderings are employed. GIF [Ghosh et al. 2020] and Exp-GAN [Lee et al. 2022] make use of the FLAME morphable face model [Li et al. 2017] in a similar vein for building controls into a GAN generator. VariTex [Bühler et al. 2021] uses the Basel Face Model [Gerig et al. 2018] to train a VAE in a similar manner. None of the above methods are unsupervised in the sense that they require the controllable variables such as camera pose and FLAME parameters as explicit input data. GIF and DiscoFaceGAN additionally require labeled lighting and texture parameters. In contrast, our generator does not require labeled data or controlled inputs.

3 OUR GENERATOR ARCHITECTURE

Figure 2a shows the architecture of our generative pipeline. An overview of its stages and their roles and design rationales is given below, with the following subsections providing details of the latent structure and the operations of these stages.

(1) Our generator first synthesizes a 3D mesh and an associated neural texture map [Thies et al. 2019] in a fixed UV parameterization. The mesh captures the identity-related shape of the head, as well as its deformation that represents facial expression, in a view-independent manner. The neural texture provides following stages with building material that is glued to the surface.
(2) Using a synthetic camera sampled by the model itself, the mesh and its neural texture are rasterized into a 2D image using a differentiable renderer. The rendering operation is a consistent mapping between the surface and pixel domains. This greatly simplifies the task of the following learned stages that process pixels — they do not need to learn to, e.g., translate in pixel space, but can instead concentrate on synthesizing a realistic image from the layout provided by the rasterizer.
(3) The rasterized result is used to condition the layers of a StyleGAN2-like foreground synthesis network that produces an RGB image and an associated alpha mask. In addition to the pixels inside the rendered mesh, this stage synthesizes complex structures such as eyeglasses and hair that can stretch far beyond the mesh in the final image. This is possible thanks to the fixed layout provided by the rasterizer, in relation to which these features can be placed.
(4) The final image is composited from the foreground and a separately-generated background image using traditional alpha blending [Bielski and Favaro 2019].

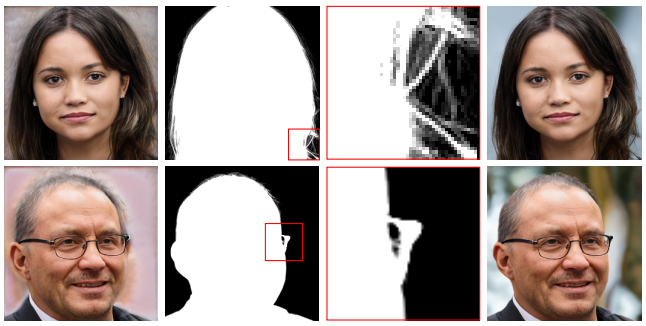
3.1 Latent Structure

Our generator takes four normally distributed latent vectors as input that aim to control different attributes of the generated images. Their wiring reflects our assumptions about the statistical dependencies observed in the data. We use the FLAME [Li et al. 2017] 3D Morphable Model (3DMM) for representing head shapes. It features orthogonal controls of overall shape (associated with identity) and expression (smile, frown, etc.) and applies the controls to a fixed-topology mesh.

As facial appearance and head shape are clearly not independent, we use an identity latent $z_{\text{ident}} \in \mathbb{R}^{512}$ to drive both the synthesis of the neural texture and the FLAME parameters that control overall head shape. We assume that facial expression is independent from identity and control its generation by another latent $z_{\text{expr}} \in \mathbb{R}^{512}$ that drives the expression parameters in FLAME. To complete the geometric part of the synthesis pipeline, we assume that the pose of the head relative to the camera, as well as other camera parameters such as field of view, are independent from identity and expression, and drive their synthesis with a third latent $z_{\text{cam}} \in \mathbb{R}^{32}$.

The synthesis of the background image is controlled by the environment latent $z_{\text{env}} \in \mathbb{R}^{512}$. While we assume that the background and illumination are mostly independent of identity, expression, and camera pose, we do note that the illumination and tonal balance need to be consistent between the foreground and background. This necessary connection is achieved by conditioning parts of the foreground network with z_{env}, as detailed below.

We emphasize that at no point do we instruct the model on how to employ the freedoms it has. There are no individual loss



(a) Foreground (b) Alpha mask (c) Inset (d) Composite

Figure 3: The foreground generator produces an RGB image (a) along with a scalar alpha mask (b). The insets (c) illustrate how features such as hair and eyeglasses are assigned fractional, high-quality coverage that allows the background to show through. The final composite (d) is obtained from the foreground and background images using traditional alpha blending. Background image not shown separately.

terms or pre-trained inference models that would indicate what effects certain degrees of freedom should have, nor do we condition the generator on a specific identity or pose. All disentanglement observed in the results is an emergent property of the unconditional learning task and the model structure.

3.2 Geometry and Texture Synthesis

A learned StyleGAN2 network $G_{\text{tex}}(z_{\text{ident}})$ produces a 32-channel 512×512 neural texture map that is later used when rendering the mesh. This neural texture passes identity-specific information as well as common positional information through the renderer to the foreground network. The FLAME parameters that control head and face shape are determined by $G_{\text{geom}}(z_{\text{ident}})$, a learned three-layer MLP. A separate latent z_{expr} affects the FLAME parameters that control expression, similarly via a learned MLP $G_{\text{expr}}(z_{\text{expr}})$.

Camera parameters are controlled by z_{cam} that also feeds into FLAME parameters that control the pose of eyes and neck. This connection is necessary, as camera position w.r.t. the subject is not independent of eyes and neck — in our training data, the subject commonly looks at the camera and is turned towards it. In total, the learned MLP $G_{\text{cam}}(z_{\text{cam}})$ outputs the eye and neck FLAME parameters, 3D camera position, 3D camera rotation, and post-perspective 2D translation and rotation parameters for the image. The post-perspective transformations are required to reproduce images that have been cropped from somewhere other than the center of the image and exhibit the consequent distortions. The 2D distance between eyes is constant in our training data, and we therefore determine the post-perspective scale so that the projected distance between vertices at the centers of the pupils is normalized.

Finally, environment latent z_{env} controls the background and light direction. A learned StyleGAN2 network $G_{\text{bg}}(z_{\text{env}})$ synthesizes a 512×512 RGB background image from which a random 256×256 crop is chosen along the horizontal midline. This randomized crop prevents the system from learning to synthesize the face or components such as hair into the background layer [Bielski and Favaro 2019]. In addition, a learned MLP $G_{\text{light}}(z_{\text{env}})$ outputs a light direction vector for the renderer.

3.3 Rendering

The fixed-function renderer rasterizes the provided mesh to produce *conditioning images* for the foreground network. These consist of texture, normal, and illumination images. For higher fidelity, all images are rendered in 1024×1024 resolution and downsampled by 4× using an approximate Gaussian kernel. We rasterize and texture map the meshes with an efficient fixed-function differentiable renderer [Laine et al. 2020].

The textured output is simply the 32-channel neural texture mapped onto the mesh with fixed UV coordinates, using mipmapping [Williams 1983] to avoid aliasing. Normals are computed in world space and interpolated across the mesh to produce the normal image. Finally, two illumination images are computed; diffuse and reflective. The diffuse component is the dot product between the mesh normal at a pixel and the light vector provided by G_{light}, whereas the reflective component is the dot product between normal and half-vector between camera and light vectors, inspired by the Phong reflection model [Phong 1975].

The foreground network additionally uses *noise images* to synthesize high-frequency detail at each resolution level, as inspired by the StyleGAN [Karras et al. 2019] architecture. With standard image-based noise, the synthesized content tends to stick to the image pixels [Karras et al. 2021], resulting in clearly visible motion artifacts. In our architecture, the noise images are provided by the renderer, and they are created by mapping a 12-channel, normally distributed random texture onto the mesh. The benefit of this approach is that the noise follows the geometry and animates naturally with the head, and thus no position-based motion artifacts are introduced in the foreground network.

Importantly, when computing mipmaps for the randomized noise texture, the data is scaled by a factor of two at each downsampling step. This retains the noise variance and prevents oblique surfaces from losing contrast due to accessing coarser mip levels.

3.4 Foreground Synthesis

The internal architecture of the foreground network is shown in Figure 2b. The design is based on the synthesis network of the original StyleGAN [Karras et al. 2019] architecture with the following modifications.

First, the style modulation mechanism is removed for all but the last three convolution layers. Instead, we use a simple image-based conditioning setup where we concatenate the conditioning images to the feature stack prior to every other convolution. The conditioning images are downsampled as needed to match the resolution of the foreground network at each step.

Second, the geometry-aware noise images are taken from the renderer instead of generated locally. These are similarly downsampled to match the feature resolution, but the variance is retained by boosting the signal magnitude according to the downsampling ratio. A different noise channel is used at each of the 12 summations in the foreground network, ensuring that the noise is unique at each step. Before each summation, the used noise channel is broadcast to the current number of feature maps and scaled by

a learned per-channel constant, similar to the original StyleGAN architecture.

Finally, the last two spatial convolution layers of the foreground network use the modulated convolution operations of StyleGAN2. The modulation input is $\boldsymbol{w}_{\text{bg}}$, i.e., the output of the mapping network of the background generator $\boldsymbol{G}_{\text{bg}}$. This enables the foreground network to adapt the overall color palette to the environment latent $\boldsymbol{z}_{\text{env}}$. Without such connection, it would not be possible to synthesize a wide range of believable foreground/background combinations.

The final layer is a 1×1 modulated convolution that narrows the output to four channels that are interpreted as RGB and alpha. To stabilize training, the alpha channel is forced to opaque for pixels that are well within the silhouette of the rasterized mesh, and to transparent far outside the silhouette. This forces at least some pixels in the composited image to originate from the foreground and some from the background, preventing early collapses to states that neglect either branch completely.

4 IMPLEMENTATION

Our system is implemented in PyTorch. We use nvdiffrast [Laine et al. 2020] as the differentiable fixed-function rasterizer, and the original implementation of FLAME [Li et al. 2017] as the morphable head model. During training, we use adaptive discriminator augmentation [Karras et al. 2020a] and therefore base our implementation on the official StyleGAN2-ADA codebase.

4.1 Network Details

When not stated otherwise, all convolutional and fully-connected layers use leaky ReLU with $\alpha = 0.2$ as the activation function.

Our geometry, expression, camera and light networks are all MLPs with three fully-connected layers. The geometry and expression networks $\boldsymbol{G}_{\text{geom}}$ and $\boldsymbol{G}_{\text{expr}}$ have 256 hidden features at each layer, and the final layer of both networks uses scaled tanh activation compress the output values to a fixed range. Most shape and expression parameters are then mapped to range $[-3, +3]$ by scaling the output of the tanh activation, except for the jaw parameters that are limited to smaller, physically plausible ranges.

The light direction network $\boldsymbol{G}_{\text{light}}$ has 32 hidden features at each layer, and the activation of the last layer is linear. The raw output is a 2D point on a plane in front of the face, which is then converted into a normalized 3D direction vector via stereographic projection. This allows the 3D light vector to obtain any direction, with the raw output $(0, 0)$ mapping to light arriving directly at the front.

Finally, the camera network $\boldsymbol{G}_{\text{cam}}$ has 32 hidden features at each layer, and tanh activation in the last layer. The raw output is scaled on a per-parameter basis to obtain values in reasonable, physically plausible ranges for camera, eye, and neck parameters.

In the foreground synthesis network, the initial learned 4×4 constant tensor has 96 feature maps, and the feature map count is kept constant throughout the network until the final 1×1 convolution layer with linear activation.

In total, the networks in our generator have 27.5M trainable parameters, which is 11% more than a standard StyleGAN2 generator network in 256×256 resolution (24.8M parameters).

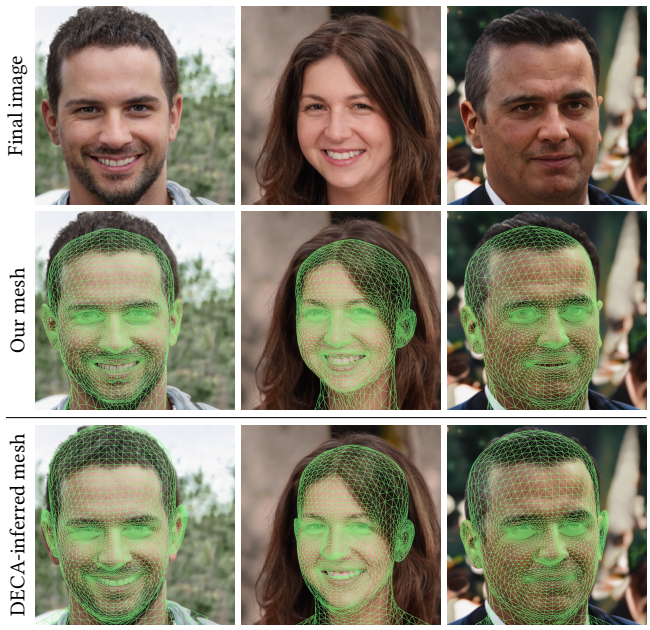


Figure 4: Top: Geometric match between the final image and the head shape and pose generated by the geometry, expression, and camera networks for three latent codes. The accurate correspondence enables responsive and intuitive control of the final image via manipulation of the 3D mesh. Bottom: Mesh inferred by DECA [Feng et al. 2021] from our rendering. The weaker correspondence (see esp. eyes and mouths) suggests that automatic geometric labeling would not be reliable in producing training targets or estimating whether the rendering follows the mesh accurately.

4.2 Training

We use the StyleGAN2-ADA training setup in the 'auto' configuration at resolution 256×256, which determines the training hyperparameters, parameter counts of the internal StyleGAN2 networks $\boldsymbol{G}_{\text{bg}}$ and $\boldsymbol{G}_{\text{tex}}$, and the discriminator network used during training. To speed up training, we modify $\boldsymbol{G}_{\text{bg}}$ from this baseline by disabling the residual skip connections and restricting the maximum feature map count to 96 throughout the network, as the task of synthesizing background is fairly trivial — this reduces the parameter count of $\boldsymbol{G}_{\text{bg}}$ by a factor of ten. For optimization, we use Adam [Kingma and Ba 2015] with parameters $\lambda = 0.0025$, $\beta_1 = 0$, $\beta_2 = 0.99$.

Our training dataset is FFHQ-U [Karras et al. 2021], the unaligned but fixed-scale version of the more common FFHQ set. The lack of alignment ensures that our learned networks are not specialized for a specific image-space position of the face, which would be detrimental to our goal of flexible control.

We trained on 4 NVIDIA V100 GPUs for 10M images, corresponding to 2.8 days of wall-clock time. Due to our higher memory consumption compared to standard StyleGAN2 training, we reduced the minibatch size to 32.

We employ standard L_2 regularization to the FLAME and camera parameters to stabilize the early stages of training. The regularization weights start high and are lowered as the training progresses to

Figure 5: Changing the 3D camera's field of view (equivalently, focal length) carries over to the rendered face but does not affect the background.

allow sufficient variation. Specifically, the regularization weights of FLAME and camera parameters are $w_{\text{FLAME}} = 10 \cdot 0.001^{\min\{x/1024,\,1\}}$ and $w_{\text{CAMERA}} = 0.1^{\min\{x/256,\,1\}}$, respectively, where x is the number of thousands of training images shown.

5 RESULTS

We now turn to the evaluation of the images produced by our model, as well as performing targeted tests to study its controllability and disentanglement. The supplemental ZIP file contains videos organized on a simple HTML page.

Overall, we find the perceptual image quality to be good. As seen in Figure 7 (page 9), the model produces a variety of identities in different poses and expressions. Notably, the background and foreground appear consistent with each other in terms of illumination and color balance, thanks to the style-based conditioning mechanism by which the background network guides the last layers of the foreground network. As shown in Accompanying Video 1, interpolation in the latent space produces pleasing results that have a distinctly different feel from the latent walks of our basis, StyleGAN2. In particular, the results are highly equivariant — i.e., the facial features and texture move with the mesh — even though our CNNs are not equivariant.

5.1 Geometric Control and Animation

As shown in Figure 4, the facial features in the final image line up very well with the shape and pose of the face model controlled by the networks G_{geom}, G_{expr}, and G_{cam}. This is a highly desirable result. As the rasterized head model and the neural texture control the synthesis of the final pixels by the foreground network, the accurate match means that changing the shape and position of the head mesh will yield corresponding near-equivariant changes in the final pixels. In other words, the features represented by the neural texture "stick" to the surface while it moves. We are aware of no prior data-driven model that produces photorealistic results and allows such fine-grained control. See the interactive editing session in Accompanying Video 4 for a demonstration.

Figure 4 also illustrates that although the geometric model only features a deforming bald head, the model comfortably synthesizes realistic hair that both reaches far outside the silhouette of the rendered model, and merges consistently with the hair inside the silhouette. We initially expected this to require helper geometry attached to the head. We also note that eyeglasses are often faithfully synthesized as transparent so that they let the background show through (Figure 3). Finally, the bottom row in Figure 4 shows that the meshes inferred by DECA [Feng et al. 2021] do not match the rendered images or their base meshes as well. In particular, eyes and mouths are often misaligned; the model also incorrectly infers a closed mouth for the leftmost sample. This calls into question the practice of treating such inference results as ground truth in evaluation or as training targets.

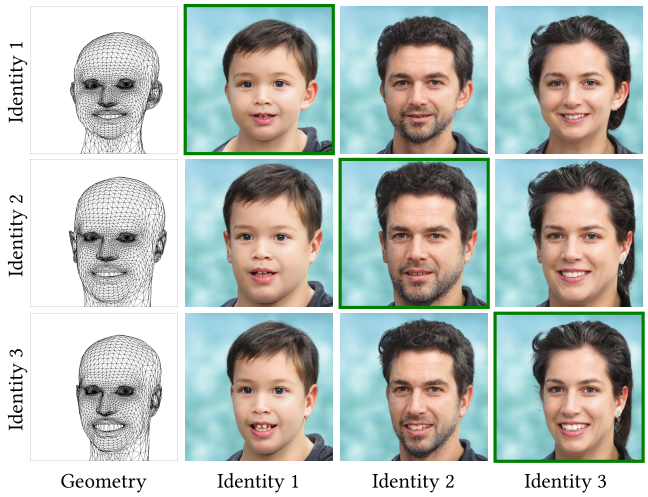


Figure 6: The identity latent z_{ident} affects both facial geometry and neural texture. Normally these are in agreement, but we can artificially use a different identity latent for the two purposes. This yields plausible but sometimes off-distribution images. Here, the "geometric" identity is kept constant in each row, and the "texture" identity in each column. The diagonal images with green borders use matching latents, whereas off-diagonal images show mixed latents. The geometry is shown on the left.

To demonstrate a use of accurate geometric control, and that our results are not inextricably tied to the FLAME head model, Accompanying Video 2 features several captured facial animations rendered using several different identity and background latents. The animations are represented as time sequences of vertex offsets relative to a base pose — *not* time varying FLAME parameters — meaning that the head shapes in the animations are not necessarily within FLAME's span. Note that this representation also forces us to use the same fixed base geometry for all the subframes, despite them having different identity latents. As seen on the video, the facial poses are represented well, with effects such as appropriately deepening wrinkles clearly visible. The only major problems are found in teeth that do not move realistically and in extreme jaw poses that are not present in the training data. The animations are sourced from Ranjan et al. [2018] and processed to remove jitter by L_1 smoothing.

As could be expected, the foreground synthesis network cannot properly handle geometric setups that are not well represented in the training data. Such failure cases are illustrated in Figure 8 (page 9). A more complete training set should mitigate such problems but might also make training more difficult.

5.2 Disentanglement

The design rationale for our generator architecture is to enable it to easily disentangle several key factors in the data. We find this succeeds well for all geometry-related parameters, and mostly succeeds for background separation.

5.2.1 Identity vs. background. Figure 10 (page 10) demonstrates the disentanglement between identity and background. Holding the background latent constant (columns) results in similar lighting conditions for the foreground, while keeping the identity latent constant (rows) results in a highly similar face structure and expression. We notice that the model does entangle features related to ethnicity, such as skin tone and hair color, with illumination. This behavior is similar to the original StyleGAN models — both illumination and skin and hair tones can be modeled with adjustments to color balance. Fully disentangling these factors may require specific constraints.

5.2.2 Expression vs. pose. Figure 9 (page 10) demonstrates, for a single identity latent, how a constant camera latent z_{cam} (rows) yields a fixed camera pose and identity despite the changing expression, while a constant expression latent z_{expr} (columns) results in a fixed identity and expression across camera poses; in other words, these three factors are not entangled with each other.

5.2.3 Field of view. Figure 5 demonstrates the disentangled control over field of view. With the latents fixed, the camera is changed from telephoto-like (small field of view) to wide angle-like (large field of view). The results show how the telephoto's apparent lack of perspective turns into the familiarly exaggerated one of the wide angle, with the identity and other properties remaining fixed.

5.2.4 Texture and head shape. Finally, we study the disentanglement of the neural texture and face geometry in Figure 6, where we intentionally break their connection at inference time. This is of interest because their generation is controlled by the same latent code z_{ident}, and hence it would, in principle, be possible that only valid combinations sampled by the model would result in reasonable images. This turns out not to be the case. First, with the camera, background and expression latents fixed, three identity latents are sampled. When fed to G_{geom} and G_{tex}, this results in three separate face geometries (leftmost column) and neural textures (not shown). In each row of the figure, a fixed face geometry is texture mapped with the three different neural textures inside the generator. The images on the diagonal are thus produced from matched and the off-diagonal ones from mismatched geometry-texture combinations. As can be seen, matching the head shape with the neural texture always appears natural, whereas the other combinations can result in images with unusual proportions. Despite this, the part of appearance and identity encoded in the texture remains constant over the columns, indicating disentanglement between shape and texture, and generalization outside the data manifold.

5.3 Illumination

We do not fully reach our goal of interpretable control over the dominant lighting direction, but see partial success in a limited effect from the lighting controls (see Accompanying Video 3). Two factors make us believe a solution is not far. First, the model *does* reliably make use of the shading map in an interpretable, albeit modest, manner; second, during the research we have seen several models where the effect is much more pronounced. An example is shown in Accompanying Video 3. Though we have thus far not been able to make this behavior emerge consistently, we believe the issue is that of subtle balancing.

5.4 Quantitative Evaluation and Image Quality

Numerically, our model yields a Fréchet Inception Distance (FID) of 12.4. This is significantly higher (worse) than the FID of 5.14 obtained by a vanilla StyleGAN2 model trained on the same 256×256 FFHQ-U dataset. To shed light on the apparent inconsistency between poor FID and the observed high quality of the images, we measure precision (P) and recall (R) as proposed by Kynkäänniemi et al. [2019]. Precision is an estimate of the fraction of generated images that match training data, i.e., are of high quality, whereas recall measures the degree to which the model covers the variation in the training data. We measure a significantly higher precision of $P = 0.82$ compared to StyleGAN2's $P = 0.63$, indicating that a large fraction of the results are indeed good. Conversely, we observe a large drop in recall ($R = 0.25$ vs. StyleGAN2's $R = 0.38$), indicating that our model leaves a larger fraction of the variation in the dataset unmodeled. This explains the increase in FID.

This tendency to drop modes is not surprising. While we have purposefully engineered strong inductive biases towards generating a single person against a background, a large fraction of the images in FFHQ-U have auxiliary foreground features, such as other persons, microphones, and the like; our model does not have good tools for modeling them, and the GAN objective allows the model to ignore them. In addition, we observe a lack of ethnic diversity in the generated images. While we have not conducted a quantitative evaluation, we suspect this can be traced back to imbalances in the dataset.

While our results are more equivariant than those of vanilla StyleGAN2, we do observe some features sticking to the pixel grid instead of moving with the head. Teeth are problematic in this sense, and locks of hair synthesized outside the rasterized mesh tend to stay fixed to the pixel coordinates more than those inside the silhouette. This limitation could likely be eliminated by moving to the alias-free architecture of StyleGAN3 [Karras et al. 2021]. Our choice of 256×256 output resolution and the non-equivariant StyleGAN2 is mostly motivated by limited computational resources.

We provide additional qualitative comparisons between our work, DiscoFaceGAN [Deng et al. 2020], GIF [Ghosh et al. 2020] and Exp-GAN [Lee et al. 2022] in Accompanying Video 5. As shown in the video, DiscoFaceGAN entangles background with identity and pose, and shows sticking artifacts inside the silhouette. GIF exhibits background flicker and identity variation in animation and pose changes, and ExpGAN flickers due to an intermediate resolution of 64×64 pixels.

6 CONCLUSIONS

We have shown that a carefully architected combination of fixed-function graphics primitives, convolutional neural networks, and MLPs can learn an accurately controllable facial synthesis model in an unsupervised manner, i.e., purely through inductive bias instead

of labeled data and variety of forcing mechanisms. In the context of face synthesis, we are interested in exploring the design space of combinations of learned and fixed-function components to enable further interpretable controls (e.g., illumination, reflectance) without sacrificing quality. Of course, the controllability of our model comes together with specialization; although we have not tried, our model is unlikely to be directly suitable for images other than faces. The design principles themselves should be applicable to other domains, though.

There is potential for future work in exploring similar approaches for other specific domains, as well as in engineering similar control mechanisms for more general scenes. This requires careful consideration of the fixed-function components and, in particular, the latent structure.

ACKNOWLEDGMENTS

We thank Tero Karras, Pauli Kemppinen, Tuomas Kynkäänniemi and Erik Härkönen for discussions and feedback. This work was partially supported by the European Research Council (ERC Consolidator Grant 866435), and made use of computational resources provided by the Aalto Science-IT project and the Finnish IT Center for Science (CSC).

REFERENCES

Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. 2017. CVAE-GAN: Fine-grained image generation through asymmetric training. In *Proc. ICCV*.

Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. 2018. Towards open-set identity preserving face synthesis. In *Proc. CVPR*.

Adam Bielski and Paolo Favaro. 2019. Emergence of object segmentation in perturbed generative models. In *Proc. NeurIPS*.

Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).

Marcel C. Bühler, Abhimitra Meka, Gengyan Li, Thabo Beeler, and Otmar Hilliges. 2021. Varitex: Variational neural face textures. In *Proc. ICCV*. 13890–13899.

Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. 2022. Efficient geometry-aware 3D generative adversarial networks. In *Proc. CVPR*.

Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. 2019. Learning to predict 3D objects with an interpolation-based differentiable renderer. In *Proc. NeurIPS*.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Proc. NeurIPS*.

Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. 2020. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In *Proc. CVPR*.

Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. 2021. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Trans. Graph.* 40, 4 (2021).

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. 2022. Get3D: A generative model of high quality 3D textured shapes learned from images. *arXiv preprint arXiv:2209.11163* (2022).

Baris Gecer, Binod Bhattarai, Josef Kittler, and Tae-Kyun Kim. 2018. Semi-supervised adversarial learning to generate photorealistic face images of new identities from 3D morphable model. In *Proc. ECCV*.

Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. 2018. Morphable face models — an open framework. In *Proc. IEEE Int. Conf. on Automatic Face & Gesture Recognition*.

Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael J Black, and Timo Bolkart. 2020. GIF: Generative interpretable faces. In *Proc. 3DV*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. In *Proc. NeurIPS*.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering interpretable GAN controls. In *Proc. NeurIPS*.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020a. Training generative adversarial networks with limited data. In *Proc. NeurIPS*.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. In *Proc. NeurIPS*.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020b. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*.

Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. 2021. Exploiting spatial dimensions of latent in GAN for real-time image editing. In *Proc. CVPR*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. ICLR*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proc. ICLR*.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2019. Improved precision and recall metric for assessing generative models. In *Proc. NeurIPS*.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. 2020. Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.* 39, 6 (2020).

Yeonkyeong Lee, Taeho Choi, Hyunsung Go, Hyunjoon Lee, Sunghyun Cho, and Junho Kim. 2022. Exp-GAN: 3D-aware facial image generation with expression control. In *Proc. ACCV*.

Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. *ACM Trans. Graph.* 36, 6 (2017).

Jinpeng Lin, Hao Yang, Dong Chen, Ming Zeng, Fang Wen, and Lu Yuan. 2019. Face parsing with RoI tanh-warping. In *Proc. CVPR*.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proc. ICCV*.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. HoloGAN: Unsupervised learning of 3D representations from natural images. In *Proc. ICCV*.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 2009. A 3D face model for pose and illumination invariant face recognition. In *Proc. IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*.

Bui Tuong Phong. 1975. Illumination for computer generated pictures. *Commun. ACM* 18, 6 (1975).

Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. 2018. Generating 3D faces using convolutional mesh autoencoders. In *Proc. ECCV*.

Axel Sauer, Katja Schwarz, and Andreas Geiger. 2022. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In *Proc. SIGGRAPH*.

Yujun Shen, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaoou Tang. 2018. FaceID-GAN: Learning a symmetry three-player GAN for identity-preserving face synthesis. In *Proc. CVPR*.

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020. StyleRig: Rigging StyleGAN for 3D control over portrait images. In *Proc. CVPR*.

Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.* 38, 4 (2019).

Luan Tran, Xi Yin, and Xiaoming Liu. 2017. Disentangled representation learning GAN for pose-invariant face recognition. In *Proc. CVPR*.

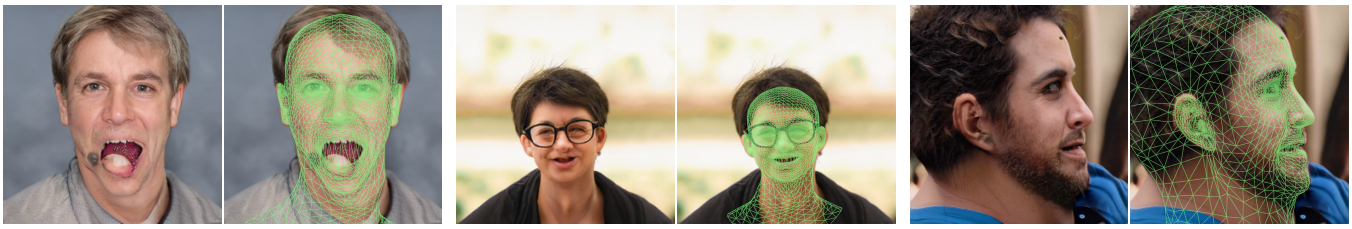
Lance Williams. 1983. Pyramidal Parametrics. *Comput. Graph. (proc. SIGGRAPH)* 17, 3 (1983).

Yu Yang, Hakan Bilen, Qiran Zou, Wing Yin Cheung, and Xiangyang Ji. 2022. Learning foreground-background segmentation from improved layered GANs. In *Proc. WACV*.

Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. 2017. Towards large-pose face frontalization in the wild. In *Proc. ICCV*.

Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G. Schwing, and Alex Colburn. 2022. Generative multiplane images: Making a 2D GAN 3D-aware. In *Proc. ECCV*.

Figure 7: A curated collection of output images with random latents. Curation ratio approx. 10%.



Extreme jaw pose Far distance Side view

Figure 8: Failure cases. In extreme geometric setups that are not well represented in the training data, the foreground network fails to synthesize a believable image based on the mesh.

Figure 9: Consistency of identity and expression across different poses.

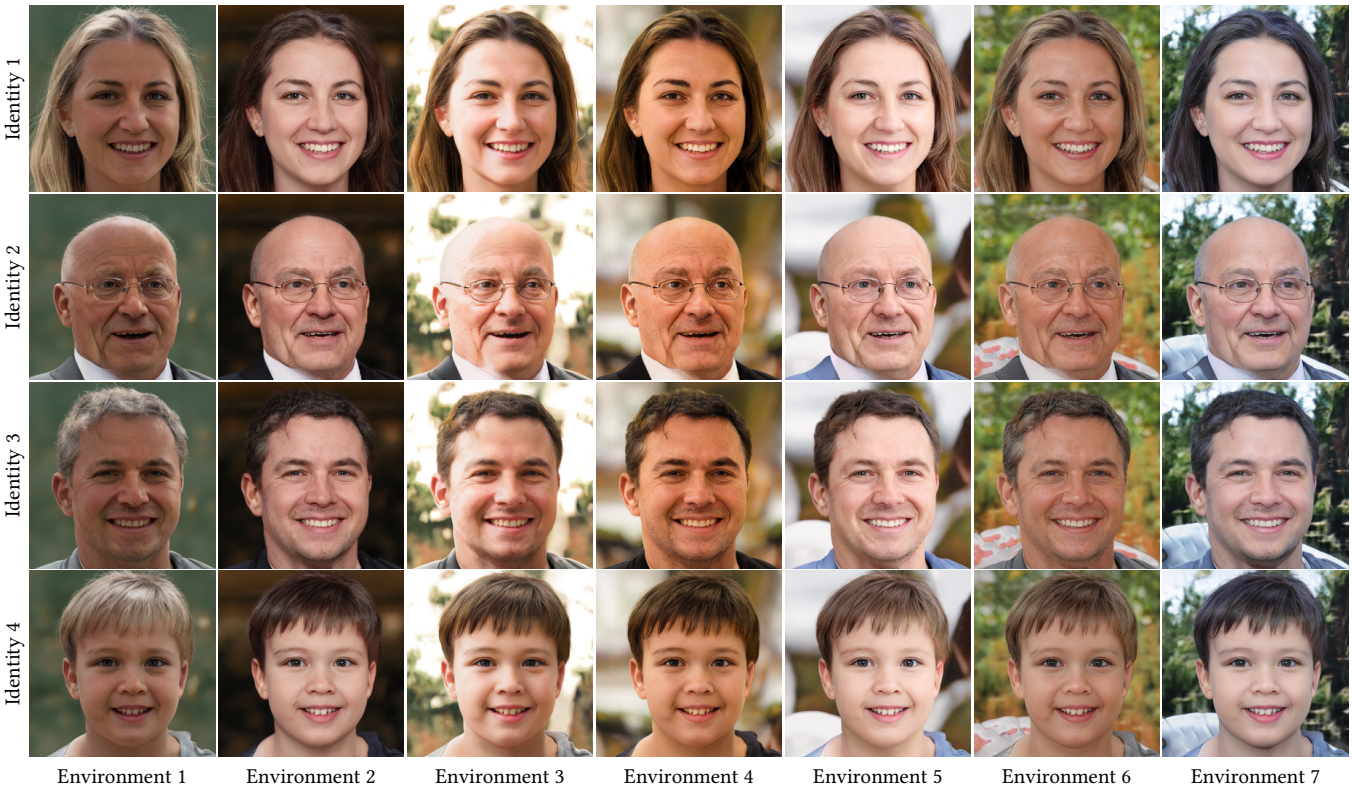


Figure 10: Consistency of identity across different environments.