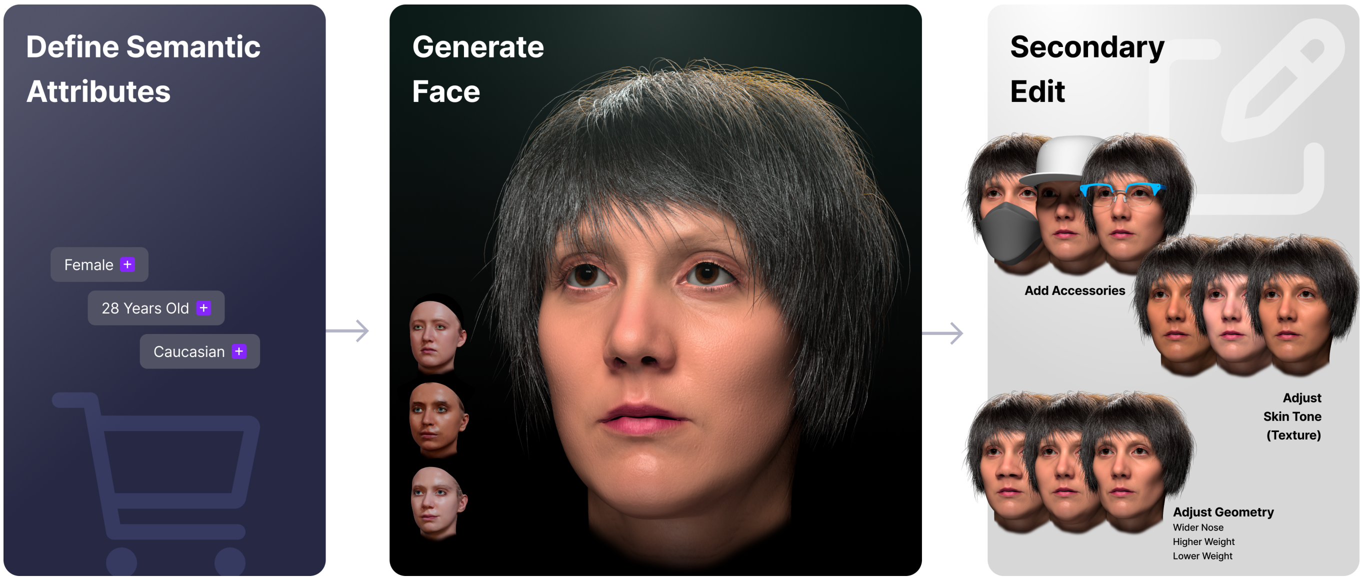
Bringing Diversity from Diffusion Models to Semantic-Guided Face Asset Generation

ANONYMOUS AUTHOR(S)



(a) (b) (c)

Fig. 1. We propose a high-quality, novel semantic controllable 3D face assets generator. This allows users to create customized avatars with a full spectrum of assets for realistic rendering. (a) User-defined semantic labels. (b) Avatars generated and rendered using all the assets. Users can generate multiple avatars and select their preferred subject. (c) Our model also supports texture space and geometry space editing, and offers a handcrafted accessory database (e.g., hairstyles, hats, glasses) for users to choose from.

Digital modeling and reconstruction of human faces serve various applications. However, its availability is often hindered by the requirements of data capturing devices, manual labor, and suitable actors. This situation restricts the diversity, expressiveness, and control over the resulting models. This work aims to demonstrate that a semantically controllable generative network can provide enhanced control over the digital face modeling process. To enhance diversity beyond the limited human faces scanned in a controlled setting, we introduce a novel data generation pipeline that creates a high-quality 3D face database using a pre-trained diffusion model. Our proposed normalization module converts synthesized data from the diffusion model into high-quality scanned data. Using the 44,000 face models we obtained, we further developed an efficient GAN-based generator. This generator accepts semantic attributes as input, and generates geometry and albedo. It also allows continuous post-editing of attributes in the latent space. Our asset refinement component subsequently creates physically-based facial assets. We introduce a comprehensive system designed for creating and editing high-quality face assets. Our proposed model has undergone extensive experiment, comparison and evaluation. We also integrate everything into a web-based interactive tool. We aim to make this tool publicly available with the release of the paper.

CCS Concepts: • **Computing methodologies** → **Computer graphics**.

Additional Key Words and Phrases: Human Face Generation, Semantic Face Manipulation, Text-Driven Generation, Generative Adversarial Networks

ACM Reference Format:
Anonymous Author(s). 2025. Bringing Diversity from Diffusion Models to Semantic-Guided Face Asset Generation . *ACM Trans. Graph.* 1, 1 (April 2025), 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM 0730-0301/2025/4-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Creating realistic 3D human faces has been a long-standing goal in both the digital industry and academia. Differently from a 2D portrait, 3D face assets contain detailed information on skin texture, facial geometry, and materials in order to be applicable in the physically based rendering (PBR) pipeline. The demands for them are rapidly growing in many of today's digital industries - gaming, movie, teleconference, and AR/VR, to list a few. However, traditionally, creating such an asset in modern industrial practices requires a time-consuming process from professional artists and, in the case of 3D scans, a meticulously calibrated facial capture studio. As of today, high-quality 3D face assets are still very costly to create.

Naturally, many past academia and industry, reasearch and engineering efforts have been made to automatize the creation of high quality 3D human faces, in a hope to make this technology more accessible to small studios and individuals. The scarcity of available face asset dataset, however, posts major challenges to even the current state-of-the-art face generation models. Industrial toolkits such as the MetaHuman [Fang et al. 2021], for example, are based on preset face assets that restrict the modeling capabilities to discrete data points and necessitates mannual editing with professional artist knowledge. Conventional 3D Morphable Models (3DMMs) [Booth et al. 2018; Cao et al. 2014; Li et al. 2017; Paysan et al. 2009] and learning-based methods [Li et al. 2020b; Yang et al. 2020; Zhang et al. 2023a] have two main limitations: they struggle to effectively integrate textures with underlying geometry, and their parametric spaces fail to capture the full spectrum of human facial geometry [Liu et al. 2022]. Additionally, these models lack robust semantic and attribute controls for generation.

The issue of data scarcity becomes more pronounced when considering modeling the diverse distribution of not just facial geometries, but also facial textures that encode variations of skin tone, freckles, wrinkles and even pores, etc. In DreamFace [Zhang et al. 2023a], a highly relevant work to ours, the proposed model has been shown to create 3D face assets with high quality, but we observe that the diversity of the synthesized textures are limited and biased to Asian skin due to the database bias (over 90% of FaceScape [Yang et al. 2020] is Asian), as only 618 textures are available for them to fine-tune a diffusion model, which we believe is insufficient. While editing features like tattoos and makeup can be achieved using a pre-trained texture LDM with prompts, this approach only allows for single-pass editing without precise control and does not support continuous modifications. Increasing texture diversity of synthesized digital humans is a key motivation under many recent 3D generative models that leverage the vision-language model CLIP [Radford et al. 2021] (as seen in AvatarClip [Hong et al. 2022]) or large diffusion model (as seen in DreamAvatar [Cao et al. 2023]) as guidance. As CLIP and many large diffusion models learn from in-the-wild data that is larger in size by some orders of magnitude, knowledge distillation from them towards digital human generation has been shown effective, particularly in the much increased diversity of the generated textures. In the more general domain, LucidDreamer [Liang et al. 2023] addresses this issue by introducing Interval Score Matching (ISM), which produces high-quality 3D models with more details and better multi-view consistency. However, 3D face generated from the above methods often contain artifacts and incomplete assets, and are not in a format readily usable for animation and rendering. Additionally, the diffusion process remains slow.

Our efforts in face asset generation similarly aim to address the data limitation challenge, by leveraging the expressive power of a large diffusion model to diversely synthesize these assets. However, we have added three key design layers that distinguish our framework from previous 3D face generation models and bridge the quality gap between scanned high-quality data and mass-generated synthetic data: **1)** UV textures estimated from diffusion-generated portraits is by itself incomplete and of arbitrary shading, which are adapted poorly to the PBR pipeline. We devise a novel method to process these reconstructed UV textures through texture completion and a normalization step that produces **clean, complete albedo maps** from the raw textures. **2)** We additionally take careful considerations in supporting meaningful **semantic control and editing** of the generated 3D faces. While the semantic labels are extracted from a pretrained text-guided diffusion model, we propose several designs that greatly improve our final model's ability to both control and edit its generated faces based on a certain facial attributes, when compared to the previous works that also incorporated this feature [Wu et al. 2023; Zhang et al. 2023a; Zhou et al. 2024]. The first step taken is to formulate the semantic attributes with the demographic properties: ethnicity, gender, and age. This circumvents the ambiguity of using a general text prompt and language model to interpret user inputs and provides more accurate control. The second step replaces the common practices of conditioning the attributes in generation with a **disentangled framework** based on the generative adversarial network (GAN) [Goodfellow et al. 2014a]. In this framework, a two-step adversarial training scheme, extended from [Xiang et al. 2021], is designed to learn an encoder that extracts unlabeled information (e.g. identity), and a generator that produces semantically accurate image from labels and the unlabeled information. This design not only circumvents the difficulty of inversion with a diffusion model, but also is validated by experiments (see Figure 13 and results in supplementary video) to be more effective in preserving the identity of the face image that is being edited. **3)** Last but not least, efficiency is a key factor in measuring the accessibility of a face asset generation model. While traditional approaches that scans, reconstructs and cleans up a captured face may take days to produce a complete face asset [LeGendre et al. 2018], recent AI-based methods [Liu et al. 2022; Zhang et al. 2023a; Zhou et al. 2024] have greatly accelerated this process. Controlled 3D face generation by sampling from a latent diffusion model is typically slow due to its iterative nature. By distilling information to a GAN network, our generation model achieves a drastic speed up in generating shape and albedo (0.014s vs. 40s under a single Nvidia A6000 GPU).

Our training pipeline consists of two major stages. **The first stage** leverages a image-based diffusion network and a 3D face reconstruction network to create a dataset of diverse, high-quality 3D face assets. As the direct image output of the diffusion network contains lighting and other external visual effects, albedo maps are computed from a proposed texture normalization algorithm. In addition, we apply a sanity check to ensure quality and discard any data with artifacts or mismatched labels, and convert the conditioning text used to generate the data into three controlling attributes: age, gender, and ethnicity. This process resulted in a dataset of 44k set of 3D face models. **The second stage** extends from DisUnknown [Xiang et al. 2021] to train a GAN with the processed training data from the first stage. As described, this stage produces the final model that generates 3D face assets consisting of 4k geometry, albedo, specular and displacement maps in the UV space. Thanks to training under a disentanglement objective, the flexible model can either create 3D assets from an attribute description ("generate a 40 years-old, Hispanic male"), or perform inversion on a given image and subsequently edit and reconstruct a 3D asset from the image, based on an attribute description ("turn this face photo into a 3D face of a 40 years-old, Hispanic male, without losing their identity").

We summarize our contributions as follow:

- We've introduced a comprehensive, practical and novel framework for generating high-quality face assets. This system uses user-defined semantics and attributes to create PBR-based face assets, including base geometry, albedo, specular, and displacement maps, as well as secondary assets such as eyeballs, teeth, and gums. The system also allows for post-generation editing in both geometry and texture, while preserving identity. The generated face avatars can be seamlessly integrated into downstream applications for rendering and animation. Additionally, we've developed an interactive web UI for users to explore these features.
- We have developed a large, high-quality 3D face database containing $44K$ albedo/geometry pairs, complete with age, gender, and ethnicity labels. This database exemplifies an effective way to use a pre-trained diffusion model in an industry production pipeline.
- We also tackled the challenging problem of domain transfer with unbalanced data amounts in two domains. Our texture normalization framework transfers 44,000 unconstrained images into a domain containing 200 images. This allows us to combine the diversity of one domain with the quality from another. This could inspire research in this field.

2 RELATED WORK

2.1 3D Face Morphable Model

The 3D Morphable Model (3DMM), as the core component of conventional 3D face generation, was first introduced in [Blanz and Vetter 1999] as a compact representation of face models in parametric space. Since then, it has been extensively applied in face recognition [Paysan et al. 2009], face reconstruction [Bas et al. 2016; Gecer et al. 2019; Thies et al. 2016], and avatar creation [Ghafourzadeh et al. 2020; Li et al. 2020c; Nagano et al. 2018]. A comprehensive overview of 3DMM is provided in [Egger et al. 2020]. Efforts have been made to improve the expressiveness, quality, and parameterization capability of 3DMM over the past 20 years.

Previous work [Booth et al. 2016; Cao et al. 2014; Li et al. 2017; Paysan et al. 2009] reduced the cost and labor required for data acquisition and registration. [Paysan et al. 2009] introduced the first publicly available 3DMM, while [Booth et al. 2018] developed a more diverse linear model using approximately 10,000 scans. For expression modeling, blendshapes were introduced to a bilinear model [Cao et al. 2014; Li et al. 2017]. Deep learning methods have since enhanced 3DMM capabilities, enabling more diverse and robust representations [Abrevaya et al. 2018; Chandran et al. 2020; Dai et al. 2020; Li et al. 2020a; Ranjan et al. 2018; Smith et al. 2020]. [Li et al. 2020d; Yang et al. 2020] constructed high-quality 3DMM with pore-level resolution data, and [Yang et al. 2020] provided a large-scale textured 3D face dataset that represents geometry through both rough shapes and detailed displacement maps. [Li et al. 2020d] developed a non-linear 3DMM using high-resolution face scans, incorporating material attributes for physically-based rendering. However, challenges remain in expensive data collection and limited diversity.

When applied to 3D face generation, conventional 3DMMs such as [Booth et al. 2018; Cao et al. 2014; Li et al. 2017; Paysan et al. 2009] offer parameterized models for face modeling. However, they lack explicit control over individual attributes and struggle to effectively integrate textures with underlying geometry. This limitation restricts their usefulness in applications requiring customized human face generation. While learning-based 3DMMs like [Li et al. 2020b; Yang et al. 2020] enable joint modeling of texture and geometry, these models still cannot provide adequate semantic or attribute control during generation.

2.2 Text-to-3D Face

Advancements in text-to-3D avatar generation [Hong et al. 2022; Michel et al. 2022; Wu et al. 2023] have leveraged the vision-language model CLIP [Radford et al. 2021] to map natural language descriptions into latent spaces, enabling the generation of 2D images that are then converted into 3D avatars. While this approach allows attribute and semantic control, it often suffers from texture artifacts and inconsistent multi-view outputs. Beyond avatar generation, text-to-3D content creation in general has rapidly evolved, with recent works [Cao et al. 2023; Chen et al. 2023a; Lin et al. 2023; Metzer et al. 2023; Poole et al. 2022] excelling at generating diverse 3D models from text. These methods employ pre-trained text-to-image diffusion models as priors to guide the training of parameterized 3D models, ensuring multi-view consistency and text alignment through Score Distillation Sampling (SDS). However, SDS encourages average-seeking behaviors in the 3D representation, leading to over-smoothed results. LucidDreamer [Liang et al. 2023] addresses this issue with Interval Score Matching (ISM), improving detail quality, but their method remains computationally expensive — taking approximately 36 minutes to generate a 3D model.

Among existing methods, DreamFace [Zhang et al. 2023a] has made significant progress in generating high-quality face avatars. It separately trains a geometry and appearance model, fine-tuning a generic diffusion model with 618 textures from sources like FaceScape [Yang et al. 2020] and 3DScanStore [3DScanStore 2023]. However, several limitations exist:**1.)** Limited geometric diversity – Geometry is initialized from a 3DMM with only 100 bases. **2.)** Insufficient and biased texture data – 618 textures are inadequate for fine-tuning a diffusion model trained on vast in-the-wild datasets, and FaceScape's 90% Asian demographic introduces bias. **3.)** Decoupled geometry and texture generation – Ignoring their correlation, as highlighted in [Li et al. 2020b]. **4.)** Challenging inversion in diffusion models – Editing (e.g., tattoos, makeup) is constrained to one-pass modifications without precise control. **5.)** Lack of quality control – Outputs are inconsistent, and quality depends heavily on user prompts.

Inspired by DreamFusion [Poole et al. 2022] and LucidDreamer [Liang et al. 2023], we aim to refine generative 3D generation. While these models generate 3D faces from text, they often produce artifacts and require manual quality control, making them unsuitable for direct application. Nonetheless, pre-trained image diffusion models, trained on large datasets, naturally enhance diversity, provide annotations and offer great accessibility. Our approach harnesses the rich priors of image diffusion models to improve quality, control, and usability in 3D avatar generation.

2.3 Semantic Face Generation and Manipulation

Generative Adversarial Networks (GANs), pioneered by [Goodfellow et al. 2014b], have been well studied in the setting of semantic editing. A key challenge lies in achieving semantic and disentangled control in generative models, *e.g.*, randomly changing one specific attribute while preserving the other attributes. Efforts to interpolate in the latent space for smooth output variation have been explored by [Laine 2018; Shao et al. 2018], while recent research focuses on disentangling the latent space for semantic control [Härkönen et al. 2020; Jiang et al. 2021; Shen et al. 2020; Shen and Zhou 2020; Zheng et al. 2021]. Observations of vector arithmetic in latent space by [Radford et al. 2016; Upchurch et al. 2017] have led to unsupervised disentanglement methods. Semi-supervised methods apply principal component analysis for attribute identification [Härkönen et al. 2020], while supervised methods use labeled data for latent space factorization [Kowalski et al. 2020; Shen et al. 2020]. Alternative approaches utilize 3DMM for semantic control over pre-trained StyleGAN latent space [Tewari et al. 2020a; Wang et al. 2022] or train unsupervised data with labels from attribute classifiers [Khodadadeh et al. 2022]. Also, to transfer such controllability to real image editing, the GAN inversion methods [Abdal et al. 2019; Tewari et al. 2020b; Zhu et al. 2020] propose to map the real image into the latent space and thus can manipulate real images.

2.4 Facial Texture Synthesis

3D facial texture synthesis has evolved from traditional geometric and photometric methods to advanced neural network-based techniques. Traditional photometric methods, relying on polarized light to capture skin reflectance properties, laid the groundwork for detailed texture mapping [Debevec et al. 2000; Ghosh et al. 2011]. Innovations then made high-quality, single-shot captures possible for both facial geometry and reflectance [Lattas et al. 2022; Riviere et al. 2020].

The advent of deep learning has revolutionized texture synthesis. Neural networks are employed for photorealistic textures, combining low and high-frequency methods [Chen et al. 2019; Huynh et al. 2018; Saito et al. 2017; Yamaguchi et al. 2018]. Generative adversarial networks (GANs) have further enhanced the generation of detailed and realistic textures, using patch-based approaches and focusing on physical texture properties like diffuse and specular albedos [Dib et al. 2021; Gecer et al. 2019; Lattas et al. 2020, 2023, 2021]. Recent developments employ the denoising diffusion probabilistic models and text inputs for texture creation [Zhang et al. 2023a]. Super-resolution techniques enable the generation of high-resolution textures from lower-resolution inputs, enhancing detail of mesh without altering its geometry [Chen et al. 2023b].

These advancements underscore the goal towards both visual realism and adherence to the principles of physical-based rendering, which guarantees accurate material properties under diverse lighting conditions.

3 METHOD

Our goal is to create a 3D avatar synthesis model that uses attribute prompts (ethnicity, gender and age) to produce high-quality 3D face assets, including 3D geometry in the form of a position map and detailed textures in the forms of albedo, specular and displacement maps. To achieve our goal, we start by constructing a large, high-quality 3D face dataset using a pre-trained diffusion model. We introduce a unique framework in Section 3.1 to ensure the collected data are clean and complete with corresponding labels. Then, a dedicated generative network, as detailed in Section 3.2, is trained using these pairs of facial data to create the basic geometries and albedo textures. Lastly, we apply post-processing, as described in Section 3.3, to refine these basic assets and generate PBR assets.

3.1 Data Preparation

We leverage the rich priors of a large diffusion model by using it to synthesize a dataset of attribute-controlled faces. However, existing large diffusion model is limited to generating 2D portraits, does not provide identity control, and the generated portraits may not conform to the provided labels. This section details four stages needed to create 3D assets (shape and albedo texture) from the raw diffusion-generated 2D portraits. First, we use the pre-trained latent diffusion models [Rombach et al. 2022; Zhang et al. 2023b] to create a set of portraits based on semantic text and geometric information. Next, these portraits are projected onto the UV space associated with the topology of the underlying 3D surface, yielding an incomplete UV-space texture, and the geometries are refined with a single-image 3D reconstruction network. In the third stage, the incomplete UV-space textures are refined using a texture completion algorithm, and the underlying geometry is optimized to ensure consistency with the synthesized facial appearances. Lastly, we introduce a novel normalization network, trained on data from two domains - a high-quality scanned database and the generated textures dataset. It converts unconstrained textures into clean albedo, similar to the scanned data domain. This pipeline efficiently creates a high-quality, diverse, and realistic 3D facial avatar dataset, including $44K$ subjects, suitable for subsequent generator training. The data generation pipeline is illustrated in Figure 2.

3.1.1 Portraits Generation using Diffusion Models. We utilize pre-trained checkpoints from Stable Diffusion v1.5 [Rombach et al. 2022] and ControlNet [Zhang et al. 2023b] (pretrained on normal maps) to generate portraits with precise control over semantic facial attributes and geometric information. To avoid ambiguity caused by the language model and ensure precise control, we define three major demographic attributes that contribute most to human appearances and shapes: ethnicity, gender, and age. We also include generic prompts for illumination and quality control, like resolution and framing. For example, "East Asian female age 20" is one set of demographic attribute combinations. Age groups span from 15 to 75 years old. For gender, we use the labels male, female, and unisex. We also consider ethnicity/race and geographic location, resulting in 14 distinct categories: *East Asian, Southeast Asian, Middle Eastern, Caucasian, Germanic, Celtic, Slavic, Romance, Australian, Native Americans, Aboriginal, Pacific Islander, and African.* To provide guidance to the generation, the pretrained ControlNet is used

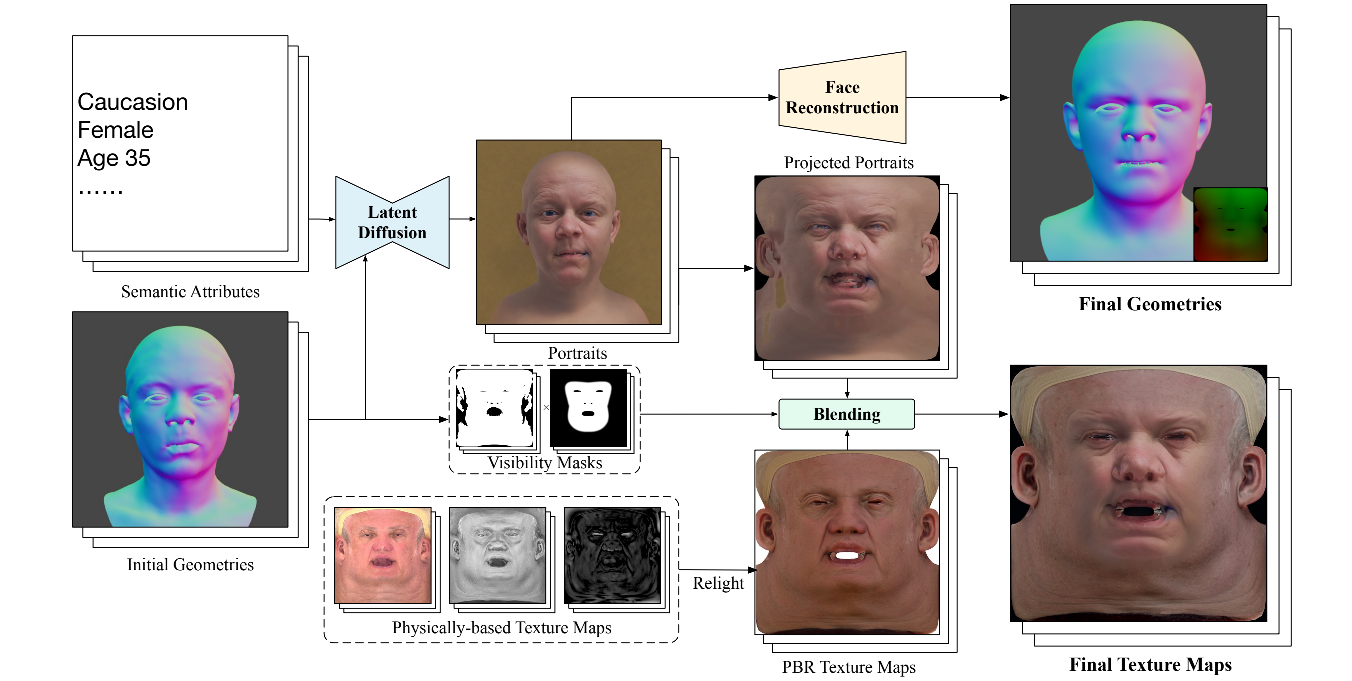


Fig. 2. Overview of the proposed dataset preparation method: Portraits are initially synthesized using a latent diffusion model that is conditioned by semantic facial attributes and frontal view normal maps. A pre-trained face reconstruction model is then applied to these portraits to extract modified geometries in the form of position maps. Textures are completed by blending the projected portraits with physically-based rendered texture maps from the scanning database.

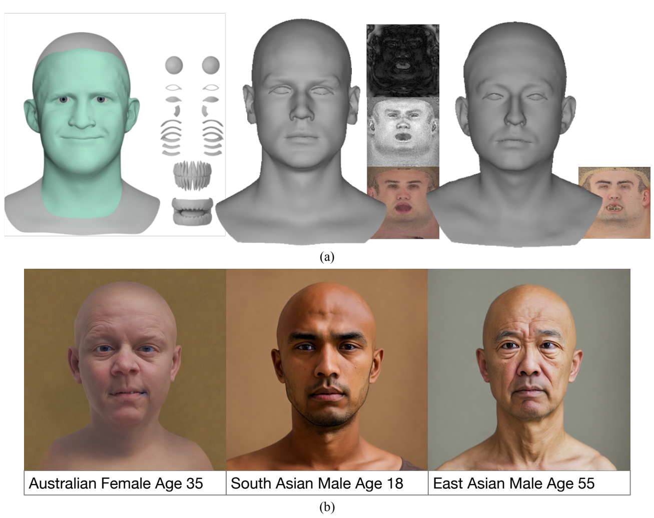


Fig. 3. Two types of 3D human face data. (a) From left to right: the template model used for registering all data resources, sample of Light Stage data, and Triplegangers data. (b) Samples of semantic attributes and the resulting 2D portraits from LDM.

to condition the diffusion process on a normal map, rendered at the front view from a 3D face randomly sampled from a scanned dataset. 3D geometry is obtained by using a single-view face reconstruction model, a variant of ReFA [Liu et al. 2022], on the 2D portraits. In total, 65k portraits with 3D geometry are created from these steps. Ethnicity is uniformly sampled from 14 categories. As for gender, the distribution consists of 45% male, 45% female, and 10% unisex. Age is uniformly sampled from the following groups: 15, 18, 20, 22, 25, 28, 30, 35, 40, 45, 55, 65, and 75.

3.1.2 Texture Completion. With the corresponding geometries, the 2D portraits can be directly resampled to the corresponding UV space. Figure 2 illustrates the projected portraits. However, since only the frontal part of the face is visible, the missing regions in the boundary of the UV maps need to be completed. We first search for the nearest neighbor with the highest similarity to the projected portrait in a synthesized texture database, using Peak Signal-to-Noise Ratio (PSNR) as our measure. This database is derived from our scanned database, and we use it to find a reference to fill missing regions. It consists of UV-space images rendered using geometry and textures from the scanned database, combined with randomized lighting, as demonstrated in the PBR texture maps shown in Figure 2. The complete UV-space texture is then obtained by blending the projected portraits and texture maps with the pyramid blending algorithm [Burt and Adelson 1983]. Finally, a sanity check is applied to filter out results with artifacts to maintain high data quality. The filtering step keeps approximately 44K UV textures from the original 65K portraits. However, these textures still contain baked-in lighting, which are not part of our desired clean albedo. Therefore, a further normalization step is proposed in the next section.

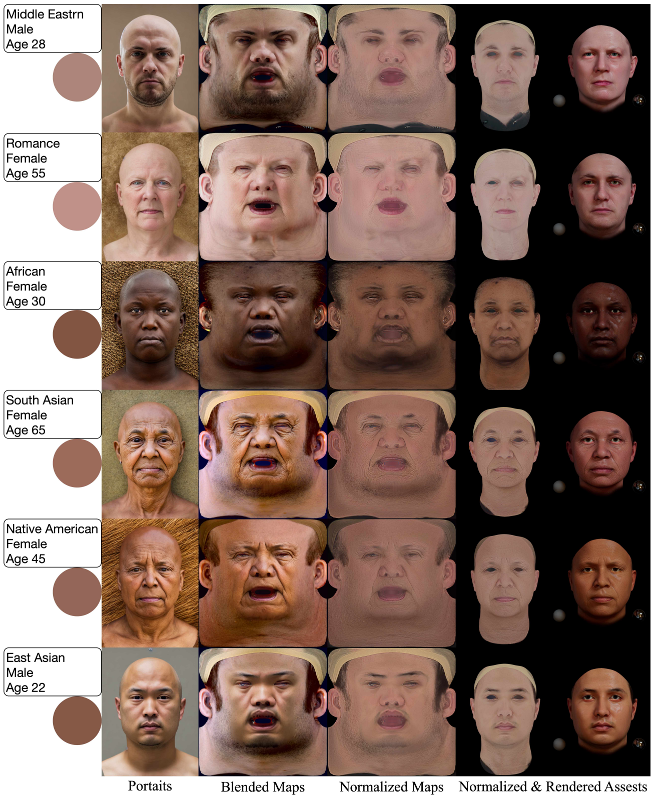


Fig. 4. Examples of training data. From left to right in each example are: the input attribute (semantic and skin tone guide), portrait, map before normalization, normalized texture map, images rendered wo/w the post-processing. (Section 3.1.3).

3.1.3 Texture Data Normalization. The goal of this step is to remove lighting and other external visual effects, such as makeup, facial hair, glasses, and shadows, from the UV textures created in the last step. The normalized texture can be considered as clean albedo map that is a part of the output 3D face asset. While FFHQ-UV [Bai et al. 2023] attempts to normalize this data into clean albedo in UV form, it struggles to fully preserve identities and attributes. We pose texture normalization in this context as a domain transfer problem from the source domain, which consists of synthetic textures under different lighting as shown in Figure 3 (b), to the target domain, which consists of clean albedo as shown in Figure 3 (a). Since the target domain contains only collected 200 identities while the source domain containing around 44K identities, a convolutional image-to-image translation model directly trained on the whole image would face a high risk of overfitting. The following text describes several designs to mitigate this issue.

First, we introduce an assumption that lighting affects appearance of the face approximately uniformly in a reasonably-sized patch. Therefore we divide an input image into a 64×64 grid of patches and assign a spatially varying factor θ to each grid corners to account for factors that are independent of albedo. θ is bilinearly interpolated in the interior of each patch. Under this formulation, image translation is formulated as a function $f(r, g, b; \theta)$ parameterized by an mlp network that takes the image as input and translates each pixel independently based on the spatially-varying factor θ. However, the image translation is preceded by a patch parameter estimation network, a convolutional network that takes the image as input and computes θ at patch corners. Effects such as subsurface scattering and inaccuracies due to the non-physical nature of the input images can also be handled by the spatial variation of θ and do not need to be considered separately.

In addition, different combinations of albedo and lighting can result in the same observed color, so without knowing the true lighting, the albedo is ambiguous. To remove this ambiguity, we compute a skin color by taking the average pixel value over manually selected flat regions of the input face (cheeks and forehead), which is equivalent to a masked average of the image $C(x)$. This value is then provided to the patch parameter estimation network explicitly. Together, the patch parameter estimation network and the mlp translation network are termed the normalization model $N(x, C(x))$, which is visualized in Figure 5.

The loss function for training the texture normalization model is constructed as:

$$\mathcal{L}_{\text{N-rec}} = \mathop{\mathbb{E}}_{x \in X_{\text{scan}}} \left[\| N(x, C(x)) - x \|_2 \right], \tag{1}$$

where X_{scan} be the set of scanned albedo texture. During training, we additionally augment the input with albedo images that do not contain lighting and optimize the network to produce identical output to the input in this situation.

The overall training procedure is shown in Figure 6. As commonly done in unpaired image-to-image translation, a patch-based discriminator is employed to ensure that the translated image is in the target domain, trained with the binary cross entropy loss. For adversarial training the distribution of skin color of the normalized synthesized textures is optimized to match that of the scanned textures, so when normalizing a synthesized texture, the skin color is drawn randomly from the set of skin colors of the scanned textures. Let D_{N} be the discriminator and X_{syn} the set of synthetic textures. The Discriminator's loss and the normalization model's adversarial loss are:

$$\mathcal{L}_{\text{N-real}} = \mathop{\mathbb{E}}_{x \in X_{\text{scan}}} \left[-\ln D_{\text{N}}(x) \right], \tag{2}$$

$$\mathcal{L}_{\text{N-fake}} = \mathop{\mathbb{E}}_{\substack{x_1 \in X_{\text{scan}} \\ x_2 \in X_{\text{syn}}}} \left[-\ln(1 - D_{\text{N}}(N(x_2, C(x_1)))) \right], \tag{3}$$

$$\mathcal{L}_{\text{N-adv}} = \mathop{\mathbb{E}}_{\substack{x_1 \in X_{\text{scan}} \\ x_2 \in X_{\text{syn}}}} \left[-\ln D_{\text{N}}(N(x_2, C(x_1))) \right]. \tag{4}$$

Additionally, we need to ensure that when normalizing a synthesized texture, the output does have the same skin color as the provided skin color:

$$\mathcal{L}_{\text{color}} = \mathop{\mathbb{E}}_{\substack{x_1 \in X_{\text{scan}} \\ x_2 \in X_{\text{syn}}}} \left[\| C(N(x_2, C(x_1))) - C(x_1) \|_2 \right]. \tag{5}$$

N and D_{N} are then trained using:

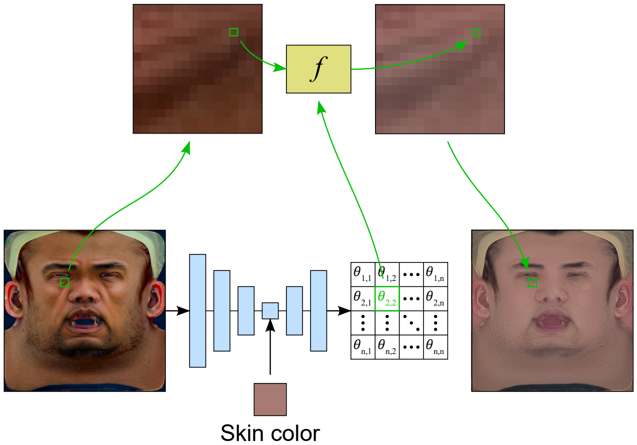


Fig. 5. Diagram of the normalization network.

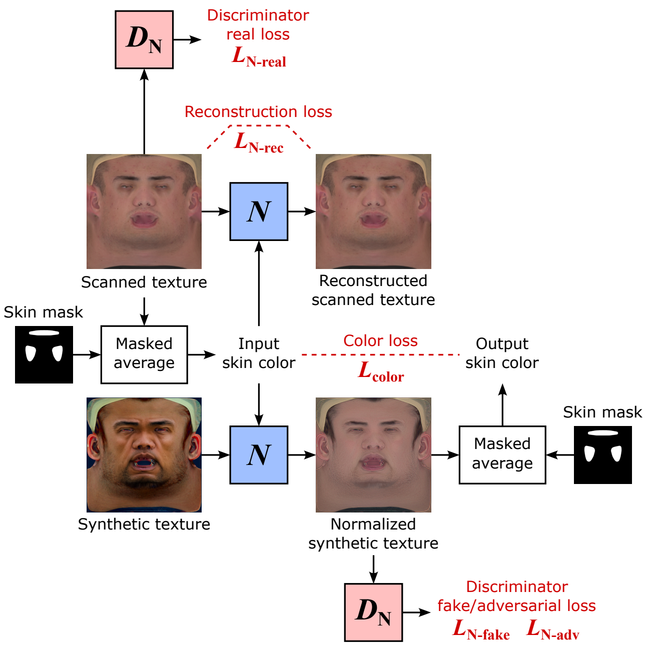


Fig. 6. Diagram of the normalization network training pipeline.

$$\min_{D_{\text{N}}} \quad \mathcal{L}_{\text{N-real}} + \mathcal{L}_{\text{N-fake}}, \tag{6}$$

$$\min_{N} \quad \lambda_{\text{N-rec}}\mathcal{L}_{\text{N-rec}} + \lambda_{\text{color}}\mathcal{L}_{\text{color}} + \lambda_{\text{N-adv}}\mathcal{L}_{\text{N-adv}}. \tag{7}$$

3.2 Base Geometry and Albedo Generation

With the prepared dataset, we train a geometry and albedo texture generation network in which each attribute of interest can be adjusted independently. In addition, the network is trained to be capable of image inversion and can disentangle identity information from attribute controls, such that it can be used as an attribute editor for a fixed identity. Our method follows the two-step approach in [Xiang et al. 2021]. The overall procedure is shown in Figure 7.

3.2.1 Learning unlabeled information. In the first step, a GAN with an autoencoder E, a generator G_1 and a discriminator D_E is designed to disentangle unlabeled information from the labels (in our case, the gender, age and ethnicity attributes). E is first used to extracts unlabeled information from the images and discards labeled information . The code computed by the encoder therefore describes information that is independent of the face attributes (e.g. identity information).

How can E be trained to disentangle information from attribute labels? If it does discard the labeled information completely, then the conditional distribution of the code, on any value of the attributes should be the same. Based on this observation, we can achieve label disentanglement by using an adversarial classifier that classifies the encoder's output by their label, and the encoder E is trained to make the classifier fail. However, while the conditional distribution of the code is learned to be the same given any label, this distribution has no closed form and cannot be easily sampled. We would like to generate new images, so this conditional distribution must be identical to some known, simple prior. So, different from [Xiang et al. 2021], the code classifier is replaced with a conditional code discriminator.

Given attribute values as conditions, the code discriminator D_{E} learns to distinguish between the prior distribution and the codes computed by the encoder. We choose the normal distribution as the prior. Let $l(x)$ be the attribute label of a training image x, and X be the training dataset, which are pairs of albedo and geometry:

$$\mathcal{L}_{\text{E-real}} = \mathop{\mathbb{E}}_{\substack{x \in X \\ z \sim p(z)}} \big[-\ln D_{\text{E}}(z, l(x)) \big], \tag{8}$$

$$\mathcal{L}_{\text{E-fake}} = \mathop{\mathbb{E}}_{x \in X} \big[-\ln(1 - D_{\text{E}}(E(x), l(x))) \big], \tag{9}$$

$$\mathcal{L}_{\text{E-adv}} = \mathop{\mathbb{E}}_{x \in X} \big[-\ln D_{\text{E}}(E(x), l(x)) \big]. \tag{10}$$

E will also need to retain all unlabeled information. This is achieved using a reconstruction loss. It is worth noting that the first-step generator will not be used as the generator in our finished texture generation model, as it only assists in training the encoder. In addition, G_1 takes the labels $l(x)$ as an input, since labeled information is removed from the codes $E(x)$. The overall reconstruction loss is:

$$\mathcal{L}_{\text{E-rec}} = \mathop{\mathbb{E}}_{x \in X} \big[\|G_1(E(x), l(x)) - x\|_2 \big]. \tag{11}$$

E, G_1 and D_{E} are then trained using:

$$\min_{D_{\text{E}}} \quad \mathcal{L}_{\text{E-real}} + \mathcal{L}_{\text{E-fake}}, \tag{12}$$

$$\min_{E, G_1} \quad \lambda_{\text{E-rec}}\mathcal{L}_{\text{E-rec}} + \lambda_{\text{E-adv}}\mathcal{L}_{\text{E-adv}}. \tag{13}$$

3.2.2 Training the label-conditioned generator. As mentioned previously, the trained $G_1(E(x), l(x))$ from step 1 does not model the marginal distribution of the attribute labels as it requires unlabeled information. In the second step, a separate conditional generator

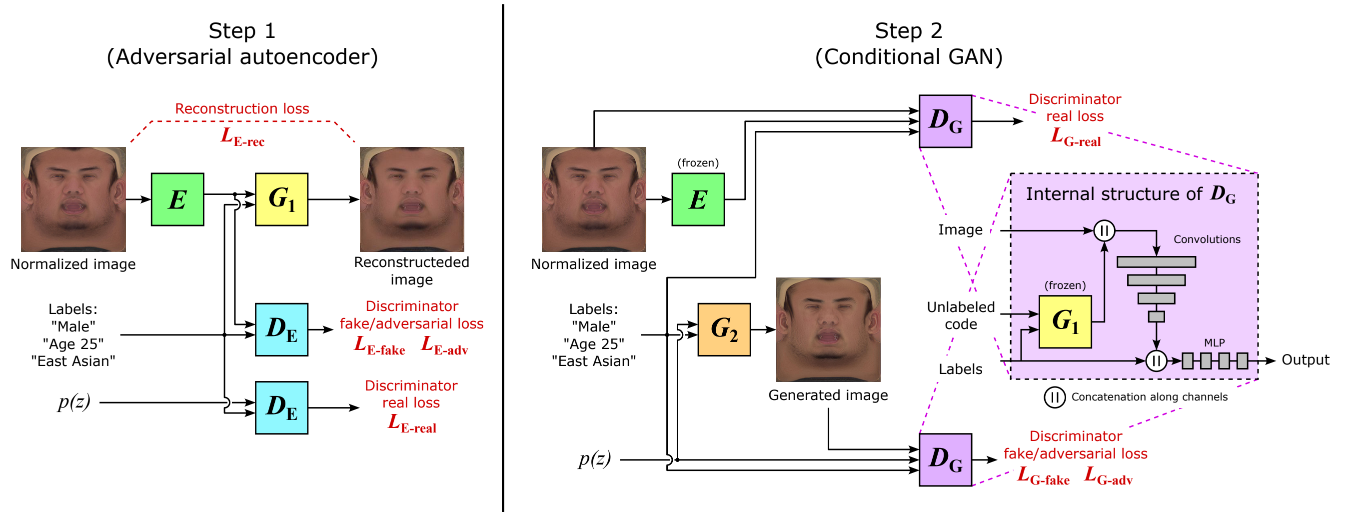


Fig. 7. Diagram of the generator training pipeline.

$G_2(z, l(x))$ is trained to generate base geometry and albedo maps from attributes and a sampled random code z. In this step, the trained autoencoder E is frozen. A new discriminator D_G receives the unlabeled code $E(x)$ as a condition and discriminates whether the generated image has preserved the unlabeled information provided by the code. Specifically, a "real" sample is a triplet consisting of a training image x, its code $E(x)$ and its label $l(x)$, while a "fake" sample is a triplet consisting of a random code z, random labels $l(x)$ which are produced by taking the labels of a random training image, and $G_2(z, l(x))$, the image generated using the code and the labels:

$$\mathcal{L}_{\text{G-real}} = \mathop{\mathbb{E}}_{x \in X} \left[-\ln D_{\text{G}}(x, E(x), l(x)) \right], \tag{14}$$

$$\mathcal{L}_{\text{G-fake}} = \mathop{\mathbb{E}}_{\substack{x \in X \\ z \sim p(z)}} \left[-\ln(1 - D_{\text{G}}(G_2(z, l(x)), z, l(x))) \right], \tag{15}$$

$$\mathcal{L}_{\text{G-adv}} = \mathop{\mathbb{E}}_{\substack{x \in X \\ z \sim p(z)}} \left[-\ln D_{\text{G}}(G_2(z, l(x)), z, l(x)) \right], \tag{16}$$

and the training procedure of the second step is a plain GAN:

$$\min_{D_{\text{G}}} \quad \mathcal{L}_{\text{G-real}} + \mathcal{L}_{\text{G-fake}}, \tag{17}$$

$$\min_{G_2} \quad \mathcal{L}_{\text{G-adv}}. \tag{18}$$

Through experiments, we noticed that although concatenating the one-hot attribute labels to the fully connected part of the discriminator worked as intended, incorporating the unlabeled code in the same way did not give satisfactory results. We speculate the difficulty lies in the observation that the unlabeled code is much longer than the attribute labels and has a strong spatial structure which the discriminator had to learn from scratch. Therefore, we help the discriminator by generating an image from the label and code it receives using the first-step generator G_1 and concatenating its output with the input image, so that the spatial structure is provided by the condition image and need not be re-learned. The detailed architecture of our discriminator is shown in the inset in Figure 7. G_1 remains frozen and is not updated along with other parameters of the discriminator.

3.3 Asset Refinement

A complete set of textures for physically-based rendering includes more than just albedo maps: it also comprises specular and displacement maps. However, our generated albedo and geometry, which are relatively coarse with 1K vertices and 1K resolution albedo maps, lack this information. In this section, we describe a refinement method that enhances our output by adding the missing assets and improving the resolution. This method includes three components.

1) Super-resolution: we train an super-resolution network [Chen et al. 2023b] to increase the 1K albedo to a 4K resolution with high-frequency details. This network not only enhances the clarity of the albedo but also recovers skin details that are lost in 1K resolution.

2) Specular and displacement maps: We train a translation network [Liang et al. 2021] that infers plausible specular and displacement maps from the 4K albedo obtained from the super-resolution network. Our model is trained using our high-quality scanned dataset. Specifically, we convert the albedo map into Lab color space and use only the L channel as input to train the displacement map network. The L channel encodes structure and illumination, while the ab channels encode color. This approach eliminates the influence of skin color, as the high-frequency geometry interpreted in displacement correlates only with shape, not skin color.

3) Secondary assets: As the final step, we integrate the missing secondary assets (such as the eyeball, teeth, and gum) and predefined Blendshapes into the obtained base mesh to prepare it for animation.

Fig. 8. Results of the single-view face reconstruction model on the portrait, (a) the initial geometry (b) the generated portrait using the initial geometry, (c) the refined geometry with single-view face reconstruction.

4 EXPERIMENTS

4.1 Implementation Details

Portraits Generation. To generate the portraits Figure 2, we use the pretrained Stable Diffusion v1.5 [Rombach et al. 2022] as base LDM model and use the ControlNet checkpoint pre-trained on normal maps [Zhang 2023] to add conditional controls for our frontal view normal map. The sampling method is DDIM with 40 time steps. Classifier-Free Guidance [Ho and Salimans 2022] scale is 9.0. The output portraits resolution from the LDM model is 1024×1024.

Single-View Face Reconstruction. In this paper, we perform single view 3D face reconstruction to update geometry. We keep the camera fixed during normal map rendering, which is used as input for the diffusion model. The resulting portraits thus have a fixed camera pose, $\mathbf{P} \in \mathbb{R}^{3\times4}$. To train the model, we rendered around $100K$ synthetic frontal view portraits under different illuminations by using assets in our 3D high quality database. We combined these with captured multi-view real data. A variant of ReFA [Liu et al. 2022] was used as the baseline to train the model, with the input modified from multi-view to single view. The camera optimization was kept fixed during training, as we had the ground truth $\mathbf{P} \in \mathbb{R}^{3\times4}$, so in each network step only position map $\mathbf{M}$ get updated. The texture inference step is also skipped. All the training of the reconstruction model is performed on NVIDIA A100 graphics cards. The network parameters are randomly initialized and are trained using the Adam optimizer for 120,000 iterations with a learning rate set to 3×10^{-4}. For the recurrent face geometry optimizer, we set the inference step to $T = 10$, the grid resolution to $r = 3$, the search radius to $c = 1$mm. An example of the refinement to the initial geometry is showed in Figure 8, the refined geometry (c) more accurately matches the portrait (b) compared to the initial geometry (a), with deeper-set eyes that are correctly positioned, a mouth that is properly sized and shaped, more correctly aligned cheekbones, and a chin that reflects the correct depth and contour, resulting in an overall head shape that aligns more closely with (b).

Texture Normalization. The Patch Parameter Estimation network is a convolutional network, consisting of alternating kernel 3, stride 1 convolutions and kernel 4, stride 2 convolutions. The number of output channels starts at 16 and doubles after each stride 2 layer. Once the spatial size reduces to 64, the skin color code is broadcast to each location, and two additional kernel 1, stride 1 layers are added. The final layer, which outputs θ, has 8 output channels. The Pixel Translation Network is an MLP with 6 layers and 64 features in the hidden layers.

To ensure that the discriminator focuses on only the lighting, its capacity is purposefully limited: it consists of an MLP with 6 layers and 64 features in the hidden layers. It takes 16×16 patches as input and first downsamples them to 4×4 so that the number of input features is $4 \times 4 \times 3 = 48$.

Generator. We adopt the StyleGAN2 [Karras et al. 2020] architecture for our generator as well as the convolution part of our discriminator. We also adopt the gradient penalty, path penalty and augmentation from its training procedure.

Performance. Our base geometry and albedo generation network generates at 0.014 seconds per subject. For 1k to 4K albedo upsampling, we trained a network [Chen et al. 2023b] on our high-quality dataset, which takes 53 seconds to complete in average. We trained two separate translation networks [Liang et al. 2021] for the specular and displacement maps and the whole translation takes 71 seconds in average. Utilizing these three types of networks, production-quality face assets for an identity can be created in less than 3 minutes on one Nvidia A6000 GPU.

4.2 Results

Qualitative Results of Generation. Figure 9 shows full assets of 16 identities along with rendered images, created from 8 distinct semantic labels. For each row, the geometry, diffuse albedo map, specular map, and displacement map are created from a set of attributes and rendered under three different lighting conditions. In each row, we display two randomly generated results that follow the same input semantic attributes, yet represent different identities. These results illustrate the diversity, quality, and effective semantic control of our model.

Texture normalization. Figure 4 showcases the direct output of our normalization network. It transfers the UV-space texture with illumination baked-in into albedo textures while preserving the identity of the input texture. We can further edit the skin tone as shown in Figure 14. For a quantitative evaluation, we calculate the Brightness Symmetry Error (BS Error) introduced by [Bai et al. 2023] of different datasets as shown in Table 1. For the Facescape data, we processed it using a method similar to [Li et al. 2020b] to obtain texture data with the same topology. For FFHQ, we employed a 3DMM fitting method to obtain partial textures from the images and then applied the blending described in Section 3.1.2 to complete the textures around the central facial area.

Attribute Control. Figure 11 shows independent control of age and gender in both texture and geometry. In each block, gender varies within each column and age varies within each row, while all other attributes remain fixed. Along the gender axis, we can see that a masculine face generally displays sharper, more pronounced features, including thicker eyebrows. Along the age axis, the most notable difference is the prominence of wrinkles.

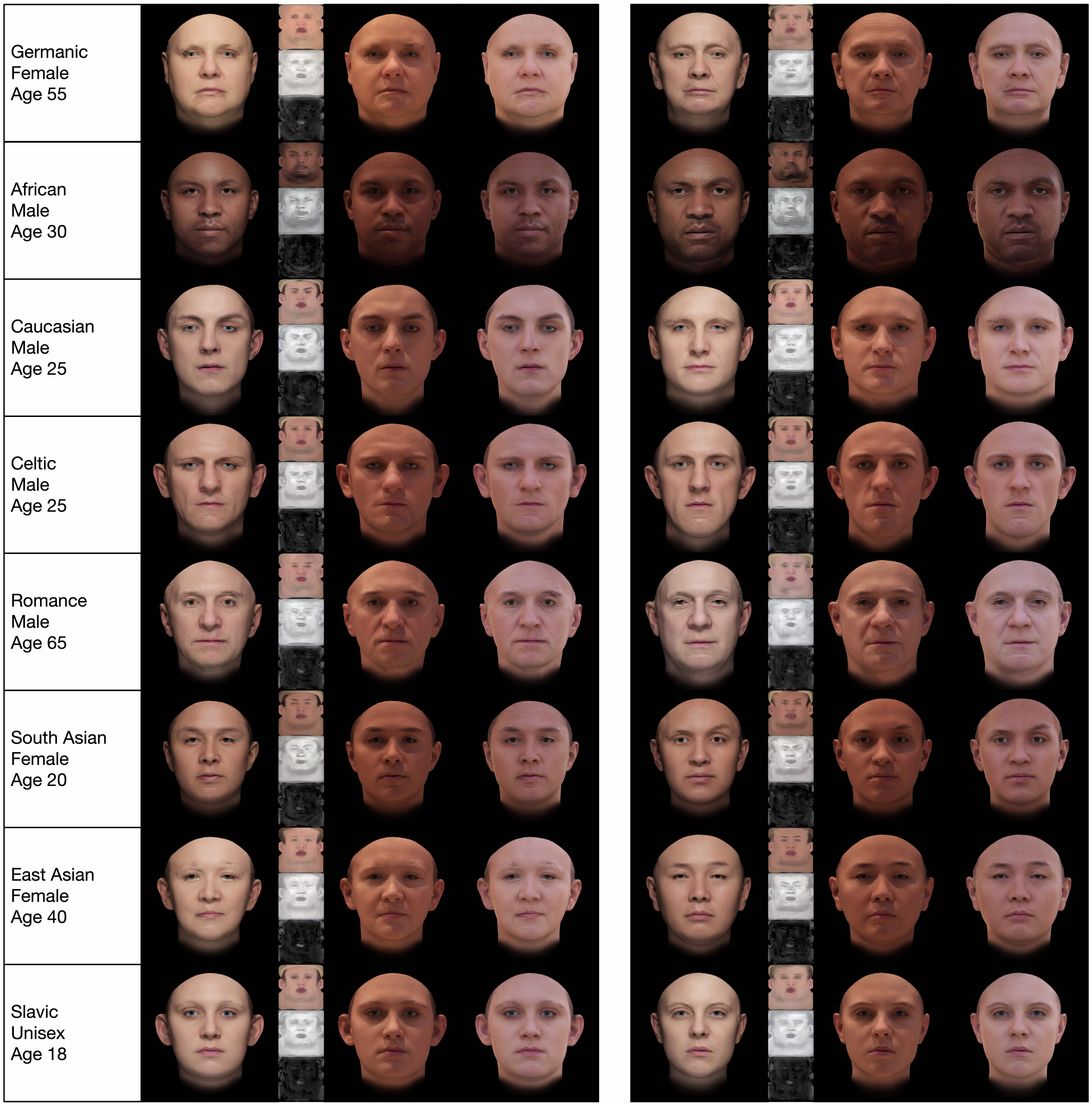


Fig. 9. Results of generated face models. For the same semantic input in each row, we generate two face models. For each of the generated face model, we show semantic labels, rendering results, PBR assets generated by the method, and rendering results under other two different lighting conditions.

Table 1. Quantitative evaluation on the illumination of the proposed UV-texture dataset in terms of BS Error, where * denotes the dataset which is captured under controlled conditions. Ours refers to FFHQ processed by proposed normalization method to match Light Stage data.

Method	Facescape*	Light Stage*	FFHQ	Ours
BS Error $\downarrow$	15.2595	4.8279	28.0723	5.8109

We also evaluate the accuracy of attribute control quantitatively. For this purpose, we train a classifier for age, gender and ethnicity, independent from the generator training process. The classifier is trained on 85% of normalized data and tested on the other 15%. We then generate a large number of samples using random codes and attribute labels, and classify these generated samples using the classifier. If the generator controls the attributes accurately, the performance of the classifier should be similar on the test data and on the generated samples. The accuracy (average of all classes) is

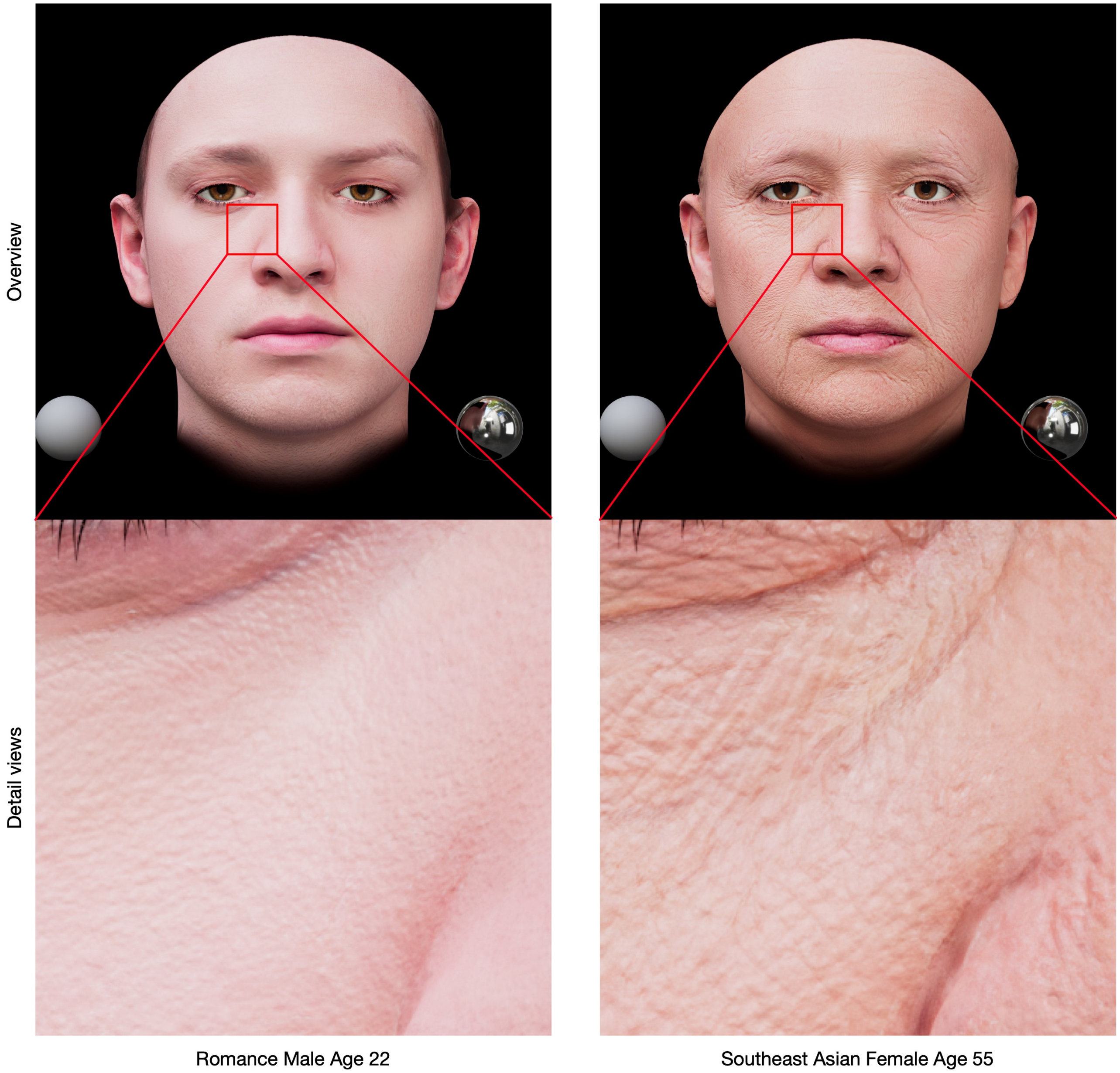


Fig. 10. Zoom-in views of generated face models. Two different subjects are shown. Top: full facial renderings with highlighted regions. Bottom: close-ups showing details of highlighted regions.

shown in Table 2, and the confusion matrix is shown in Figure 15. The performance of the classifier on the dataset and the generated samples are similar, and where they differ, the accuracy is generally higher on the generated samples.

To compare the effectiveness of attribute control between our trained generator and a generic diffusion model [Rombach et al. 2022], we perform age editing on the same subject using both methods, as shown in Figure 13. The portraits generated by the diffusion model, which is the same as the one used in portraits generation, are processed through our data preparation pipeline to obtain final renderings. While diffusion models can achieve age modification through prompt manipulation, our generator produces significantly smoother transitions across age levels. In contrast, diffusion-based results have abrupt feature changes between adjacent slices in the figure. *A more detailed comparison of transition smoothness can be found in the supplementary video.*

Figure 14 shows independent control of skin color. In particular, we show the variation of skin color within two subjects, conditioned

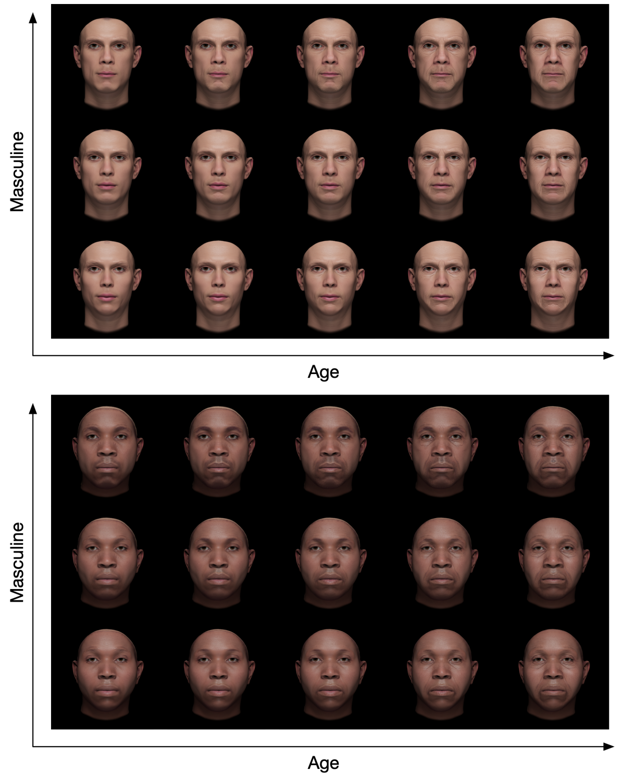


Fig. 11. Independent control of age and gender, both textures and geometries are interpolated in two dimension (age and gender). Each row varies gender from feminine to masculine, and each column increases age progressively.

Table 2. Classification accuracy on the dataset and generated samples and their difference.

	race	gender	age
dataset	60.09%	84.72%	32.66%
generated	75.88%	91.93%	49.90%
difference	+15.79%	+7.21%	+17.24%

on their respective ethnicity. We calculate the skin color from each unnormalized texture using the method in [Thong et al. 2023] and for each ethnicity the distribution of skin colors is modeled by a normal distribution. The figure presents five skin color samples of each subject, arranged from bright to dark.

Inversion and Editing. Figure 12 shows image inversion and editing. Images are first projected into the latent space of the generator by finding the unlabeled code and labeled attributes that generates the closest image using optimization. Edited images are then generated by modifying the labeled attributes while keeping the unlabeled code unchanged.

Additionally, since all generated facial geometries share the same topology with our high-quality dataset, generic facial geometric

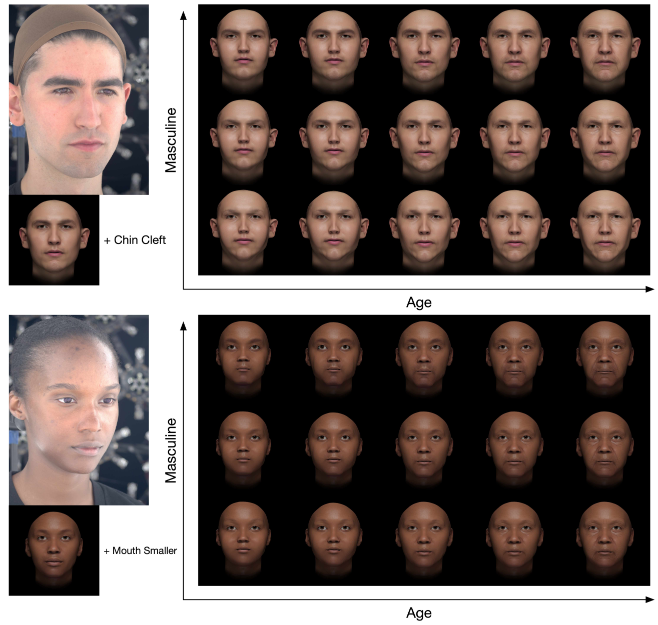


Fig. 12. Examples of GAN inversion and editing of facial features. Each subject is inverted from a real photo, then modified on the geometry—adding a chin cleft (top) or reducing mouth size (bottom)—followed by age and gender editing in the latent space of the genrator.

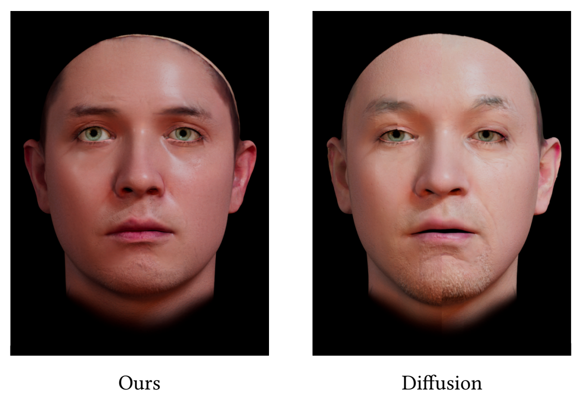


Fig. 13. Age editing comparison between our method and text-prompt-based editing using Stable Diffusion [Rombach et al. 2022]. For each method, 20 ages ranging from young to old are uniformly sampled. Each generated face contributes a vertical slice.

attributes editing such as add aging features Figure 11 can be done with some predefined geometry offsets, and blendshapes can be applied to animate the facial assets. For animation examples, please refer to the video.

Comparison. We access the accuracy of attribute control of our generator using CLIP score. We first build a text description of the generated image data for our semantic attributes. We utilize the same ethnicity, gender, and age groups as the attributes we defined for generation to construct descriptors. Following the strategy

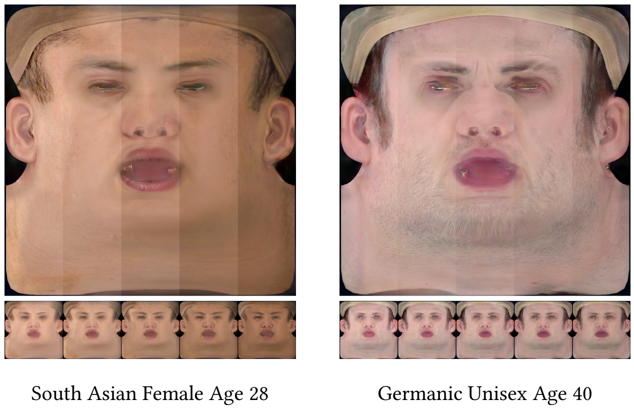


Fig. 14. Skin color editing on two subjects. For each subject, skin tone is modified from bright to dark while keeping identity and facial features consistent.

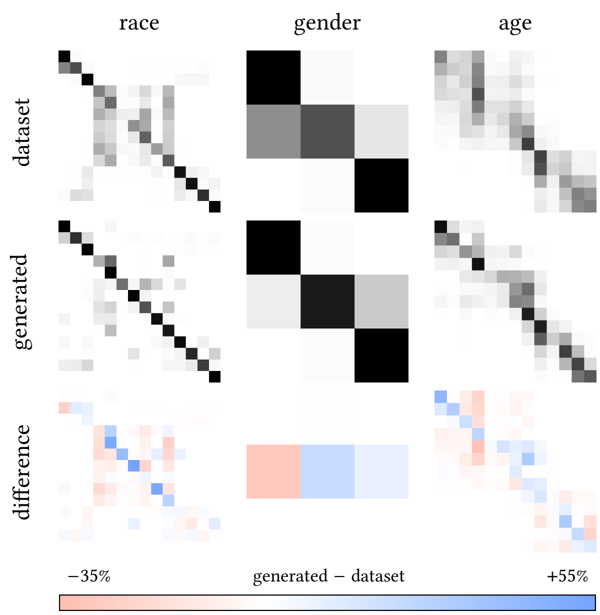


Fig. 15. Classification confusion matrix on the dataset and generated samples and their difference.

of [Zhang et al. 2023a], we form text input with the phrase "the realistic face of [DESCRIPTOR]". After acquiring 5 sets of PBR assets for each descriptor, we render 2500 test images. To calculate the score of Describe3D [Wu et al. 2023], we synthesize facial images by generating the descriptors with the rules mentioned in their work and used the same phrase as the input into CILP model. Subsequently, we compute the average CLIP score from the ViT-B/16 and ViT-L/14 models. The result, shown in Table 3, highlights effectiveness of our method in achieving semantic coherence between the generated images and their corresponding text descriptions.

Figure 16 compares our system to DreamFace [Zhang et al. 2023a]. Our system generates real albedo, which reflects true skin color, whereas DreamFace only produces texture under evenly distributed illumination. Also, our generated face model exhibits greater diversity and more closely aligns with the user description as the training dataset of DreamFace mainly consists of Asian people. Compared to general text-to-3D methods like DreamFusion [Poole et al. 2022], LucidDreamer [Liang et al. 2023] and TRELLIS [Xiang et al. 2024], our method produces avatars with impressive detail and production-level textures. DreamFusion creates clear structures, but misses finer skin details. LucidDreamer and TRELLIS outputs vivid, detailed and realistic faces, however with strong artifacts.

Table 3. Comparison of Methods

Method	CLIP Score ↑
DreamFace [Zhang et al. 2023a]	0.291 ± 0.020
Describe3D [Wu et al. 2023]	0.284 ± 0.054
UltrAvatar [Zhou et al. 2024]	0.301 ± 0.023
Our Method	**0.316 ± 0.042**

Figure 17 shows the results of texture normalization compared to FFHQ-UV [Bai et al. 2023]. Our normalization results are closer to the target albedo domain, with pure skin color and no lighting baked-in (e.g., highlights). However, the normalized UV generated by FFHQ-UV still retains some of the original illumination, which cannot be used for high-end PBR-base shader as albedo.

User Study. We conduct a user study for the output of avatar generation methods among our proposed method, DreamFace [Zhang et al. 2023a] and Describe3D [Wu et al. 2023]. 87 participants rate images synthesized using the corresponding methods in two aspects: description consistency and photorealism. The outcomes in Figure 18 showcase our proposed method consistently outperforming others in terms of description consistency and photorealism.

Interactive Avatar Creation System. To facilitate validating the on-the-fly interactivity of the proposed method, a web application using browser rendering technologies is created. Figure 19 demonstrates the attributes editing function enabled by our method after creation. Users are presented with options to specify "General Facial Structure" attributes such as age, ethnicity, and gender, and "Detailed Facial Structure" features including the shape of the face, nose, ears, chin, and more. Users can see the results in the viewport and rotate the camera and light to inspect details of the generated result. From the technical side, the application employs a front-end system for rendering and a back-end system for providing services and hosting the web pages. The front-end system uses the Unity game engine [Unity Technologies 2022] and is built against the WebGL platform. All the generation requests regarding changes to age, ethnicity, and gender are sent back to the back-end for processing and returned to the front-end as asset files once ready. The mesh object, albedo map, specular map, and displacement map are the major art assets dynamically generated by our models and saved on the cloud storage associated with an ID in our database for later reference. Changes to other facial feature settings are handled locally on the browser, and no new assets need to be generated by the server, so no significant delay is incurred at this stage to ensure an interactive editing experience. During local interactive modification,

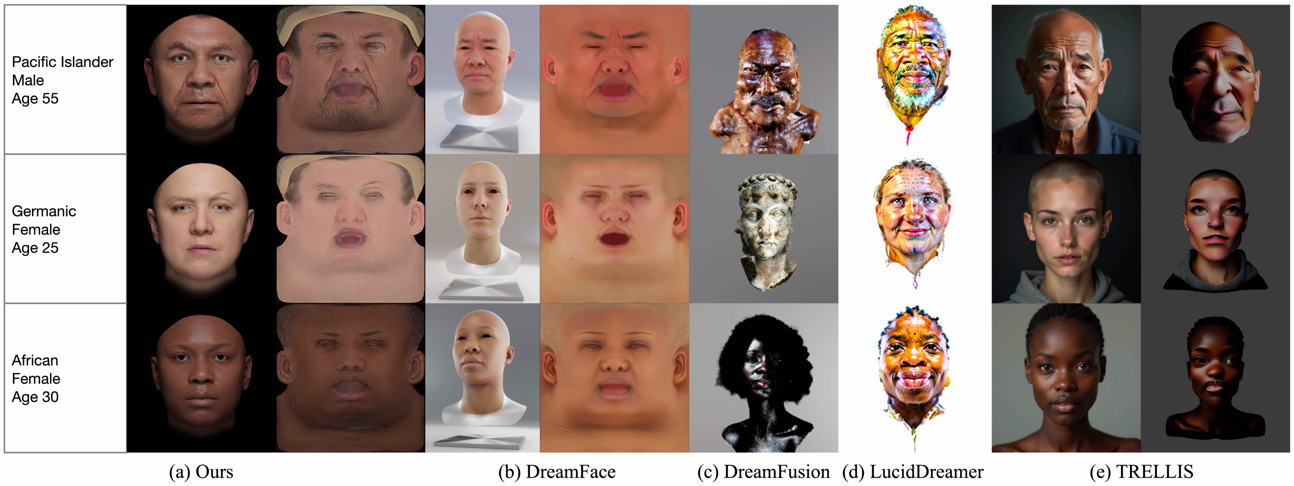


Fig. 16. Comparison of our semantic face asset generation of (a) proposed method against (b) DreamFace[Zhang et al. 2023a] (c) DreamFusion[Poole et al. 2022] (d) LucidDreamer[Liang et al. 2023] (e) TRELLIS[Xiang et al. 2024]

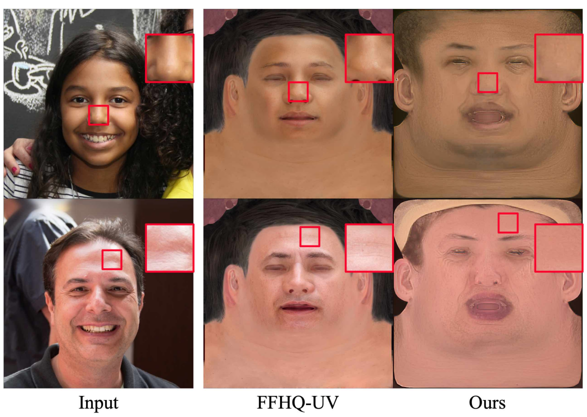


Fig. 17. Comparison of texture normalization against FFHQ-UV[Bai et al. 2023]. No highlights are baked into our results.

related assets, such as eye color textures and geometry offset presets, are hosted by the back-end server statically and are only loaded on demand. Changes to these interactively modifiable features are recorded in the database to make the result reproducible. The back-end system is implemented using ASP.NET Core [Microsoft 2024] and is modularized, so the system's capabilities can be expanded with ease. *Please check our supplementary materials for the demo video recordings and details.*

5 LIMITATIONS

While we have made considerable progress towards a production-quality system for semantic-guided generation PBR face asset, there are aspects of our system that can be improved upon.

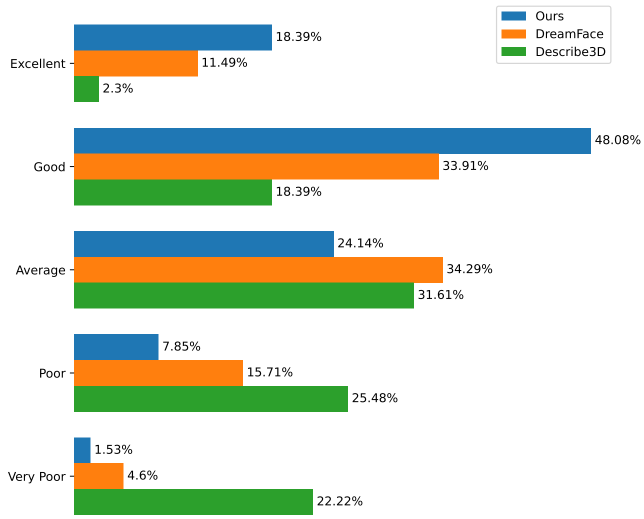
By sourcing our training data from diffusion models, we expanded the diversity of our training data greatly. However, our method only works for texture maps, as there is currently no controllable high-quality large-scale generative model for face geometry. So, while we can refine the geometry with a single-view face reconstruction method, the diversity of the initial geometry is still limited by what we have in the scanned dataset.

Furthermore, the invisible parts on the UV texture projected from the frontal view portraits, generated by the diffusion model, need to be supplemented by the scanned dataset. This requires a well-distributed scan data. If the generated portrait has no similar labels in the scanned data, for instance, a dark skin texture may not be accurately completed if there's no dark skin in the scanned database. Therefore, we aim to explore a multi-view consistent texture synthesis method that can produce a complete texture without the need for inpainting.

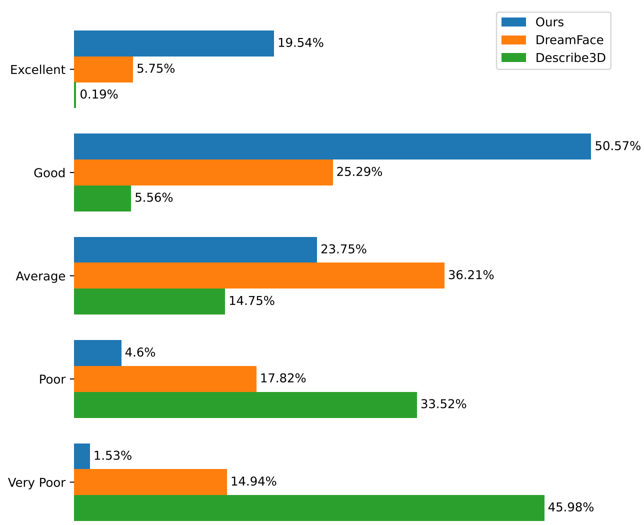
Since pre-trained diffusion models are used to create the training data, our generation model may inherit any potential biases present in models like Stable Diffusion. Although we ensure that each attribute has balanced training data through manually defined attributes and distributions in data preparation, it is worth discussing how our method might still inherit issues if the pre-trained models do not express certain attributes diversely. Additionally, due to the high quality of the generated faces, misuse and privacy management are also significant challenges.

6 CONCLUSION

We have introduced a novel 3D face generative model that allows for semantic control and creates high-quality face models, which is unprecedented in 3DMM of human face. The main contributions that realize our model is a specifically designed normalization network and a disentangled generator that are able to utilize not only the high-quality scanned models but also in-the-wild face models that are vast in quantity and semantic information. Experiments have demonstrated that our model can effectively normalize faces from arbitrary lighting conditions, generate novel face and perform



(a) Description Consistency



(b) Photorealism

Fig. 18. Comparison of different methods across various categories

attribute manipulations on the generated 3D face in multiple semantic directions. We believe that the progress our system achieves has great potential use in many applications, including VFX production, customized digital avatars and the generation of synthetic training data for other fundamental computer vision research.

Fig. 19. Interface showcasing the avatar creation web application. Users setup features to generate and refine the facial structure and features of the avatar through an interactive and intuitive graphical representation.

REFERENCES

3DScanStore. 2023. 3DScanStore. https://www.3dscanstore.com [Online; accessed 24-January-2024].

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?. In *ICCV 2019*. 4431–4440.

Victoria Fernández Abrevaya, Stefanie Wuhrer, and Edmond Boyer. 2018. Multilinear Autoencoder for 3D Face Model Learning. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, 1–9. https://doi.org/10.1109/WACV.2018.00007

Haoran Bai, Di Kang, Haoxian Zhang, Jinshan Pan, and Linchao Bao. 2023. FFHQ-UV: Normalized Facial UV-Texture Dataset for 3D Face Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 362–371.

Anil Bas, William A. P. Smith, Timo Bolkart, and Stefanie Wuhrer. 2016. Fitting a 3D Morphable Model to Edges: A Comparison Between Hard and Soft Correspondences. In *ACCV 2016 Workshops*, Vol. 10117. 377–391.

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999*, Warren N. Waggenspack (Ed.). ACM, 187–194. https://dl.acm.org/citation.cfm?id=311556

James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. 2018. Large Scale 3D Morphable Models. *Int. J. Comput. Vis.* 126, 2-4 (2018), 233–254.

James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. 2016. A 3D Morphable Model Learnt from 10, 000 Faces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 5543–5552. https://doi.org/10.1109/CVPR.2016.598

Peter J Burt and Edward H Adelson. 1983. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (ToG)* 2, 4 (1983), 217–236.

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Trans. Vis. Comput. Graph.* 20, 3 (2014), 413–425. https://doi.org/10.1109/TVCG.2013.249

Yukang Cao, Yan-Pei Cao, Kai Han, Ying Shan, and Kwan-Yee K Wong. 2023. Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models. *arXiv preprint arXiv:2304.00916* (2023).

Prashanth Chandran, Derek Bradley, Markus H. Gross, and Thabo Beeler. 2020. Semantic Deep Face Models. In *8th International Conference on 3D Vision, 3DV 2020, Virtual Event, Japan, November 25-28, 2020*, Vitomir Struc and Francisco Gómez Fernández (Eds.). IEEE, 345–354.

Anpei Chen, Zhang Chen, Guli Zhang, Kenny Mitchell, and Jingyi Yu. 2019. Photo-realistic facial details synthesis from single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9429–9439.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023a. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22246–22256.

Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. 2023b. Dual Aggregation Transformer for Image Super-Resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12312–12321.

Hang Dai, Nick E. Pears, William A. P. Smith, and Christian Duncan. 2020. Statistical Modeling of Craniofacial Shape and Texture. *Int. J. Comput. Vis.* 128, 2 (2020), 547–571. https://doi.org/10.1007/s11263-019-01260-7

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 145–156.

Abdallah Dib, Cedric Thebault, Junghyun Ahn, Philippe-Henri Gosselin, Christian Theobalt, and Louis Chevallier. 2021. Towards high fidelity monocular face reconstruction with rich reflectance using self-supervised learning and ray tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12819–12829.

Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhöfer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 2020. 3D Morphable Face Models - Past, Present, and Future. *ACM Trans. Graph.* 39, 5 (2020), 157:1–157:38.

Zhixin Fang, Libai Cai, and Gang Wang. 2021. MetaHuman Creator The starting point of the metaverse. In *2021 International Symposium on Computer Technology and Information Science (ISCTIS)*. IEEE, 154–157.

Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. 2019. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1155–1164.

Donya Ghafourzadeh, Cyrus Rahgoshay, Sahel Fallahdoust, Andre Beauchamp, Adeline Aubame, Tiberiu Popa, and Eric Paquette. 2020. Part-Based 3D Face Morphable Model with Anthropometric Local Control. In *Proceedings of the 45th Graphics Interface Conference 2020, Toronto, ON, Canada, May 28-29, 2020*, David I. W. Levin, Fanny Chevalier, and Alec Jacobson (Eds.). Canadian Human-Computer Communications Society, 7–16.

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 1–10.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014a. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014b. Generative Adversarial Nets. In *NeurIPS 2014*. 2672–2680.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6fe43269967adbb64ec6149852b5cc3e-Abstract.html

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. 2022. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535* (2022).

Loc Huynh, Weikai Chen, Shunsuke Saito, Jun Xing, Koki Nagano, Andrew Jones, Paul Debevec, and Hao Li. 2018. Mesoscopic facial geometry inference using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8407–8416.

Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. 2021. Talk-to-Edit: Fine-Grained Facial Editing via Dialog. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 13779–13788.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8110–8119.

Siavash Khodadadeh, Shabnam Ghadar, Saeid Motiian, Wei-An Lin, Ladislau Bölöni, and Ratheesh Kalarot. 2022. Latent to Latent: A Learned Mapper for Identity Preserving Editing of Multiple Face Attributes in StyleGAN-generated Images. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*. IEEE, 3677–3685.

Marek Kowalski, Stephan J. Garbin, Virginia Estellers, Tadas Baltrusaitis, Matthew Johnson, and Jamie Shotton. 2020. CONFIG: Controllable Neural Face Image Generation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI (Lecture Notes in Computer Science, Vol. 12356)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 299–315.

Samuli Laine. 2018. Feature-Based Metrics for Exploring the Latent Space of Generative Models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=BJslDBkwG

Alexandros Lattas, Yiming Lin, Jayanth Kannan, Ekin Ozturk, Luca Filipi, Giuseppe Claudio Guarnera, Gaurav Chawla, and Abhijeet Ghosh. 2022. Practical and scalable desktop-based high-quality facial capture. In *European Conference on Computer Vision*. Springer, 522–537.

Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou. 2020. AvatarMe: Realistically Renderable 3D Facial Reconstruction" in-the-wild". In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 760–769.

Alexandros Lattas, Stylianos Moschoglou, Stylianos Ploumpis, Baris Gecer, Jiankang Deng, and Stefanos Zafeiriou. 2023. FitMe: Deep Photorealistic 3D Morphable Model Avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8629–8640.

Alexandros Lattas, Stylianos Moschoglou, Stylianos Ploumpis, Baris Gecer, Abhijeet Ghosh, and Stefanos Zafeiriou. 2021. Avatarme++: Facial shape and brdf inference with photorealistic rendering-aware gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 12 (2021), 9269–9284.

Chloe LeGendre, Kalle Bladin, Bipin Kishore, Xinglei Ren, Xueming Yu, and Paul Debevec. 2018. Efficient multispectral facial capture with monochrome cameras. In *ACM SIGGRAPH 2018 Posters*. 1–2.

Jiaman Li, Zhengfei Kuang, Yajie Zhao, Mingming He, Karl Bladin, and Hao Li. 2020c. Dynamic facial asset and rig generation from a single scan. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–18.

Jiaman Li, Zhengfei Kuang, Yajie Zhao, Mingming He, Karl Bladin, and Hao Li. 2020d. Dynamic facial asset and rig generation from a single scan. *ACM Trans. Graph.* 39, 6 (2020), 215:1–215:18.

Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xiang, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, et al. 2020b. Learning formation of physically-based face attributes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3410–3419.

Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xiang, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, and Hao Li. 2020a. Learning Formation of Physically-Based Face Attributes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 3407–3416. https://doi.org/10.1109/CVPR42600.2020.00347

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), 194:1–194:17. https://doi.org/10.1145/3130800.3130813

Jie Liang, Hui Zeng, and Lei Zhang. 2021. High-resolution photorealistic image translation in real-time: A laplacian pyramid translation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9392–9400.

Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. 2023. LucidDreamer: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching. arXiv:2311.11284 [cs.CV]

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 300–309.

Shichen Liu, Yunxuan Cai, Haiwei Chen, Yichao Zhou, and Yajie Zhao. 2022. Rapid Face Asset Acquisition with Recurrent Feature Alignment. *ACM Trans. Graph.* 41, 6, Article 214 (nov 2022), 17 pages. https://doi.org/10.1145/3550454.3555509

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12663–12673.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13492–13502.

Microsoft. 2024. ASP.NET Core. https://dotnet.microsoft.com/apps/aspnet Accessed: 2024-05-19.

Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. 2018. paGAN: real-time avatars using dynamic textures. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 2009. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009, 2-4 September 2009, Genova, Italy*. 296–301.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV]

Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *ICLR 2016*.

Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. 2018. Generating 3D Faces Using Convolutional Mesh Autoencoders. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 11207)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, 725–741. https://doi.org/10.1007/978-3-030-01219-9_43

Jérémy Riviere, Paulo Gotardo, Derek Bradley, Abhijeet Ghosh, and Thabo Beeler. 2020. Single-shot high-quality facial geometry and skin appearance capture. (2020).

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. 2017. Photorealistic facial texture inference using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5144–5153.

Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. 2018. The Riemannian Geometry of Deep Generative Models. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 315–323. https://doi.org/10.1109/CVPRW.2018.00071

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the Latent Space of GANs for Semantic Face Editing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 9240–9249. https://doi.org/10.1109/CVPR42600.2020.00926

Yujun Shen and Bolei Zhou. 2020. Closed-Form Factorization of Latent Semantics in GANs. *CoRR* abs/2007.06600 (2020). arXiv:2007.06600 https://arxiv.org/abs/2007.06600

William A. P. Smith, Alassane Seck, Hannah Dee, Bernard Tiddeman, Joshua B. Tenenbaum, and Bernhard Egger. 2020. A Morphable Face Albedo Model. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 5010–5019. https://doi.org/10.1109/CVPR42600.2020.00506

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020a. StyleRig: Rigging StyleGAN for 3D Control Over Portrait Images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 6141–6150.

Ayush Tewari, Mohamed Elgharib, Mallikarjun B. R., Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020b. PIE: portrait image embedding for semantic control. *ACM Trans. Graph.* 39, 6 (2020), 223:1–223:14.

Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *CVPR 2016*. 2387–2395.

William Thong, Przemyslaw Joniak, and Alice Xiang. 2023. Beyond Skin Tone: A Multidimensional Measure of Apparent Skin Color. In *ICCV*.

Unity Technologies. 2022. Unity Game Engine. https://unity.com/ Accessed: 2024-05-19.

Paul Upchurch, Jacob R. Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Q. Weinberger. 2017. Deep Feature Interpolation for Image Content Changes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 6090–6099.

Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. 2022. Cross-Domain and Disentangled Face Manipulation with 3D Guidance. *IEEE Transactions on Visualization and Computer Graphics* (2022).

Menghua Wu, Hao Zhu, Linjia Huang, Yiyu Zhuang, Yuanxun Lu, and Xun Cao. 2023. High-fidelity 3D face generation from natural language descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4521–4530.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3D Latents for Scalable and Versatile 3D Generation. *arXiv preprint arXiv:2412.01506* (2024).

Sitao Xiang, Yuming Gu, Pengda Xiang, Menglei Chai, Hao Li, Yajie Zhao, and Mingming He. 2021. DisUnknown: Distilling Unknown Factors for Disentanglement Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14810–14819.

Shugo Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. 2018. High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. 2020. FaceScape: a large-scale high quality 3D face Dataset and detailed riggable 3D face prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 601–610.

Lvmin Zhang. 2023. ControlNet v1.1. https://huggingface.co/lllyasviel/ControlNet-v1-1/. Accessed: 2024-05-19.

Longwen Zhang, Qiwei Qiu, Hongyang Lin, Qixuan Zhang, Cheng Shi, Wei Yang, Ye Shi, Sibei Yang, Lan Xu, and Jingyi Yu. 2023a. DreamFace: Progressive Generation of Animatable 3D Faces under Text Guidance. *arXiv preprint arXiv:2304.03117* (2023).

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023b. Adding Conditional Control to Text-to-Image Diffusion Models.

Yutong Zheng, Yu-Kai Huang, Ran Tao, Zhiqiang Shen, and Marios Savvides. 2021. Unsupervised Disentanglement of Linear-Encoded Facial Semantics. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 3917–3926.

Mingyuan Zhou, Rakib Hyder, Ziwei Xuan, and Guojun Qi. 2024. UltrAvatar: A Realistic Animatable 3D Avatar Diffusion Model with Authenticity Guided Textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1238–1248.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-Domain GAN Inversion for Real Image Editing. In *ECCV 2020*, Vol. 12362. 592–608.