3D FACE MODELING: APPLICATIONS IN GENERATIVE TASKS AND OCCLUSION-AWARE RECONSTRUCTION

By

Rahul Dey

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Doctor of Philosophy

2023

ABSTRACT

3D modeling of human faces has emerged as a widely studied field within computer vision, with applications in virtual reality, animation, medical imaging, and more, and is going to be a very promising area of research in the coming years. Specifically, 3D modeling of single view face images has been known to be a particularly challenging task because of its ill-posed nature, but it comes with a wide range of applications. Two of the most promising approaches in this regard are template-based approaches, such as the 3D morphable model (3DMM) of faces, and implicit 3D modeling approaches, such as implicit 3D-GANs. Over the years, 3DMM based approaches have improved their capability to synthesize highly controllable 3D faces and generate accurate 3D face reconstructions of faces images, while implicit 3D-GANs have been shown to generate high-fidelity 3D faces. However, even after significant advancements in these approaches, face generative tasks, such as face inpainting and controllable face generation, are still primarily performed in the 2D image space.

Faces are structured 3D objects with inherent attributes such as shape, pose and albedo, and their projection in 2D images is affected by external factors such as illumination and camera parameters. Without an explicit consideration of these factors, existing generative approaches have to implicitly model facial geometry and appearance. We contend that generative models that explicitly take these factors into account can leverage 3D priors, and more controllably and accurately generate new faces, or fill in the missing regions in face images.

Further, the ill-posed nature of reconstructing 3D models from monocular face images makes it a challenging task. This becomes even more challenging when facial occlusions such as face masks, glasses, microphones, *etc*. are involved. This highlights the need for the development of occlusion-aware 3D face reconstruction algorithms. We argue that such an algorithm should be (i) robust to occlusions of varying types, sizes, and locations; and (ii) capable of generating diverse, yet realistic solutions for the occluded parts to account for a lack of unique solution.

This thesis addresses the aforementioned challenges, by presenting the following: (i) a 3D-aware face inpainting approach that considerably improves upon 2D-based baselines, especially

under challenging conditions; (ii) a controllable 3D face generation approach that combines the capabilities of 3DMMs and implicit 3D-GANs by learning correspondence between them; and (iii) an occlusion-aware 3D face reconstruction approach that generates a diverse, yet realistic set of 3D reconstructions from a single occluded face image, with lower error on the visible face regions than the baselines.

Dedicated to my parents, whose sacrifices have enabled me to reach this far.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to the people who have supported and guided me throughout my PhD journey.

First and foremost, I would like to thank my PhD supervisor Prof. Vishnu Boddeti, for the opportunity to be a part of his team from its conception, and for his guidance, expertise, constructive criticism, and unwavering support throughout my research.

I am grateful to my thesis panel comprising of Prof. Arun Ross, Prof. Xiaoming Liu, and Dr. Felix Juefei-Xu, for serving on my committee and providing their valuable feedback, mentorship, and recommendations. Their esteemed presence in my academic journey has been a great fortune, which I will always cherish.

I would also like to thank Prof. Bernhard Egger, Dr. Tim Marks, and Dr. Ye Wang for their expertise and mentoring in the CoLa-SDF project.

I am grateful to Prof. Anil Jain, Prof. Jiayu Zhou, Prof. Hayder Radha, Prof. Sijia Liu and other esteemed faculty members of MSU for the amazing course offerings that nurtured me in the ares of computer vision, machine learning, and related fundamental fields.

I also thank the staff of the CSE graduate office for their help and support, as well as the staff of the Division of Engineering Computing Services for troubleshooting many IT and compute related issues.

I would like to thank my friends, including lab mates from the HAL lab, and other members from the computer vision and machine learning groups at MSU for their support throughout this journey, and for the many fun trips and leisure activities that made this journey much more enjoyable.

Finally, I would like to express my heartfelt gratitude towards my family - Mom, Dad, and my elder sisters Deepika and Renuka for their continued love and support, without which, I would not have been able to accomplish this.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Over the recent years, the coming together of computer graphics and computer vision have led to the emergence of powerful tools for analysis and synthesis of real world objects in 3D. Such tools, when applied to human faces, have shown promising applications in AR/VR applications such as overlaying virtual faces onto the real world, creating realistic and lifelike animations of human faces, medical imaging, face recognition and so on. Specifically for faces, template based approaches such as 3D morphable models (3DMMs) [Paysan et al., 2009, Li et al., 2017a], and deep learning based approaches such as implicit 3D models [Mildenhall et al., 2020, Chan et al., 2021, Hong et al., 2021] have emerged as very promising approaches of 3D modeling. By representing faces as linear combinations of shape and texture bases, 3DMMs can not only synthesize new 3D faces in a highly controllable way, but also reconstruct 3D models from 2D face images. Their nonlinear counterparts, commonly known as nonlinear 3DMMs [Tran and Liu, 2018, Tran et al., 2019, Feng et al., 2021, Medin et al., 2022] have further improved the expressivity of these models. On the other hand, implicit 3D models represent 3D objects using implicit functions rather than meshes or surfaces, and have been shown to be capable of modeling high fidelity faces in 3D, including the regions not modeled by 3DMMs, such as the inner mouth cavity and hair [Or-El et al., 2022, Gu et al., 2021]. Despite such advances, the applicability of 3D modeling to face generative tasks such as face inpainting, and controllable face generation and editing have not been properly explored. Further, 3D face reconstruction approaches still struggle in the presence of occlusions such as face mask, glasses, microphones, *etc*.

Face inpainting is the process of reconstructing missing or corrupted parts of an image by predicting and filling in the missing pixels based on the surrounding context. It has wide applications in face editing and restoration, face de-occlusion, and virtual try-on to name a few. Existing face inpainting approaches operate in 2D. Typically a masked face image is fed to an end-to-end autoencoder network that outputs the completed face image [Li et al., 2017b, Yu et al., 2018, Yu et al., 2019, Zheng et al., 2019a]. The limitation of these approaches is that the completed faces often

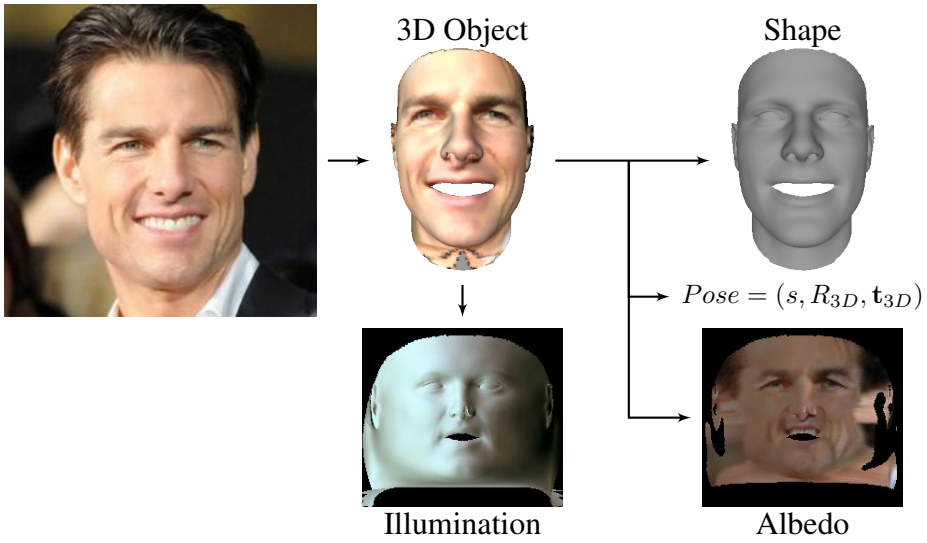


Figure 1.1 Human face is inherently a 3D structure comprised of a 3D shape, 3D pose, and albedo, and its appearance when captured in an image can be impacted by global factors like lighting, camera position, and surrounding objects.

have geometric artifacts, specially when the masks get large as shown in Fig. 1.3. We argue that this limitation can be overcome by incorporating explicit 3D priors into the model. Human faces are structured 3D objects with inherent attributes such as shape, pose and albedo. Their projection in 2D images is affected by external factors such as illumination and camera pose (see Fig. 1.1). By having to implicitly model the geometry, structure and appearance of faces, 2D-based approaches often fail to sufficiently account for them, causing such artifacts [Li et al., 2017b].

Another major generative task involving faces is the controlled generation and editing of faces. This is used to generate realistic human faces with specific attributes or characteristics and has numerous applications in gaming, virtual reality, digital advertising, forensics and law enforcement, fashion and beauty industries, *etc*. While several 2D based approaches exist [Tewari et al., 2020b, Deng et al., 2020, Tripathy et al., 2021], they often do not control attributes such as pose and hairstyle, and have limited applications wherever 3D models are required, *e.g.*, gaming and virtual reality. This necessitates the implementation of such approaches in 3D. However, while 3DMMs are highly disentangled and afford explicit control over attributes like shape, pose, albedo and illumination, they do not include the inner mouth cavity and hair, and are limited in their expressivity and realism [Feng et al., 2021, Medin et al., 2022]. Implicit 3D models, on the other

Figure 1.2 Real world face images often have occlusions caused by various objects, including face-related objects like glasses and beards, as well as unrelated objects like microphones or tools. When analyzing face images, these occlusions should be excluded from the analysis. The images are from the CelebA dataset [Liu et al., 2015].

hand, are highly expressive and generate high-fidelity 3D faces, but do not afford explicit control over the facial attributes. A high fidelity 3D generative model, with a high degree of control, needs to bring together the capabilities of both these approaches.

Another challenge in 3D modeling of 2D face images is occlusions. In-the-wild face images often come with several forms of occlusions (see Fig. 1.2). Performing monocular 3D reconstruction from occluded face images confronts several challenges: (i) *Robustness to occlusions:* The difficulty in 3D face reconstruction depends on the degree, size, shape and location of occlusions. For example, the larger the occlusion, the more difficult it gets to reconstruct an accurate face 3D model; and (ii) *Lack of unique solution:* In the presence of occlusion, it is not possible to know with certainty how the occluded face would have looked like, even if the algorithm can reconstruct a highly realistic-looking face 3D model with respect to the visible regions. In such cases, a method that can generate a distribution of diverse solutions would be preferable. Most existing methods

of monocular 3D face reconstruction do not explicitly account for occlusions, which affects their robustness to such occlusions. And the ones that do consider occlusions, do so by parsing and using only the visible parts of the face image [Song et al., 2019b, Egger et al., 2018]. This affords them some degree of robustness to occlusions, while still not accounting for the possible diversity of solutions.

In this dissertation, we study and address the aforementioned challenges. First, we present a 3D-aware face inpainting approach that incorporates explicit 3D modeling of faces, and show its effectiveness over existing approaches, specially under challenging conditions. Then, towards controllable 3D face generation, we present an approach that establishes correspondence between the parameters of a nonlinear 3DMM model and the latent space of an implicit 3D-GAN to generate highly controllable, yet high-fidelity 3D faces with explicit control over its physical attributes like shape, pose, albedo, illumination, and hairstyle. Then, we explore ways to make 3D reconstruction from monocular face images robust to occlusions, while simultaneously accounting for diversity in the occluded regions. Finally, we discuss potential future work in this area. We now provide a brief introduction of these approaches, followed by our specific contributions.

In Chapter 3, we look at the challenge with face inpainting, particularly under large variations in pose, shape, illumination, and mask sizes and locations. Existing face inpainting approaches [Yu et al., 2019, Zheng et al., 2019a] often result in poor photorealism under such conditions and often fail to preserve facial symmetry and variations in these factors while inpainting, as shown in the example of extreme face poses, illumination variations, and diverse appearances and shapes in Fig. 1.3. To this end, we present our approach called 3DFaceFill [Dey and Boddeti, 2022a], which aims to address the challenge of face de-occlusion by explicitly disentangling a face image into its 3D components. Our method completes the facial albedo in its UV representation and integrates the 3D shape, 3D pose, and illumination with the inpainted albedo to render the completed face image back. Additionally, we leverage facial symmetry in the UV representation of albedo to aid in the inpainting of symmetric occluded facial regions. Through extensive experiments across multiple datasets and challenging conditions, we demonstrate that 3DFaceFill improves

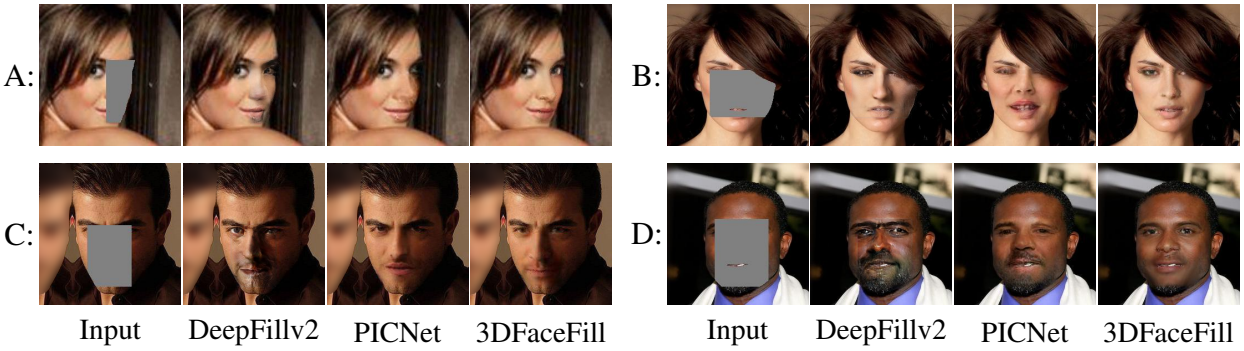


Figure 1.3 Inpainting of face images under diverse conditions by 3DFaceFill and existing approaches. By modeling the image formation process 3DFaceFill is able to generate more geometrically consistent and photorealistic completions across diverse scenarios such as non-frontal poses (A), light and dark complexions (B,D), non-uniform facial illumination (*e.g*. illumination is different on two sides of the nose in C) and in cases where the baselines tend to distort face components (*e.g*. nose in B).

face completion both quantitatively and qualitatively over the baselines [Yu et al., 2018, Zheng et al., 2019b, Li et al., 2017b, Li et al., 2020a] by as much as 4db in terms of PSNR and $\sim$25% in terms of LPIPS [Zhang et al., 2018b], a metric considered closer to human perception.

In Chapter 4, we evolve a method for controllable generation and subsequent editing of 3D faces, called CoLa-SDF. We combine the controllability of nonlinear 3DMM approaches with the high fidelity of implicit 3D-GANs by establishing correspondence between the parametric space of nonlinear 3DMM, and the latent space of 3D-GANs. Building upon the impressive photorealism and expressive 3D representation of StyleSDF [Or-El et al., 2022], CoLa-SDF adopts a similar architecture but enforces the latent space to match the interpretable and physical parameters of the nonlinear 3D morphable model MOST-GAN [Medin et al., 2022]. Through our experiments, we showcase the effectiveness of CoLa-SDF in achieving high-fidelity face synthesis and subsequent 3D manipulation with full control over the disentangled latent parameters as shown in Fig. 1.4.

While 3D modeling can lead to improved face de-occlusion, occlusions themselves present a major challenge to 3D reconstruction. This results in a chicken-and-egg problem. We tackle the issue of 3D face reconstruction in Chapter 5. We specifically focused on two aspects of the problem: *robustness* and *diversity*. Traditionally, monocular 3D reconstruction approaches, both fitting-based [Paysan et al., 2009, Li et al., 2017a, Egger et al., 2018], as well as neural network-

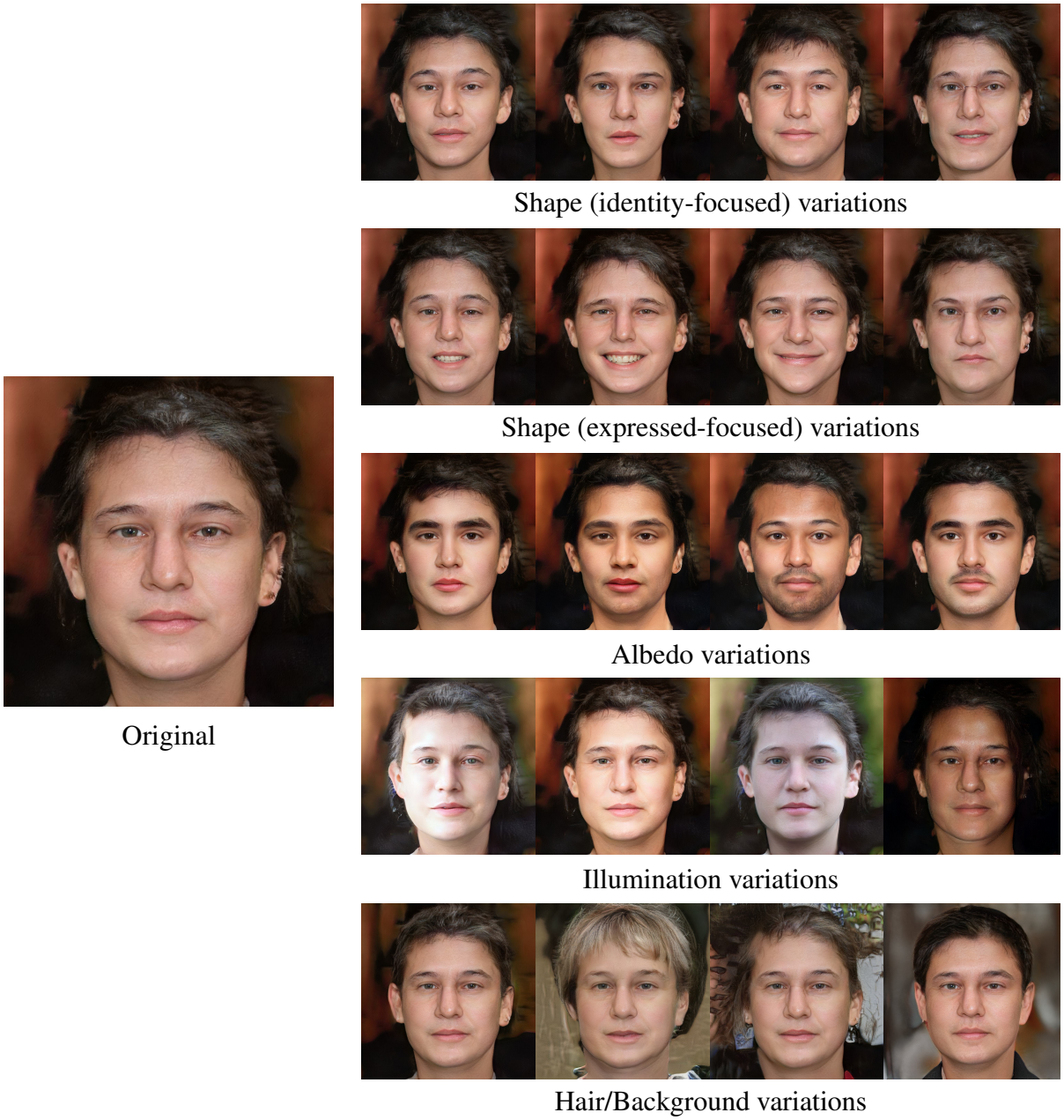


Figure 1.4 Our proposed method CoLa-SDF combines the controllability of physical attributes afforded by 3DMM-based approaches with the high-quality generative capability of implicit 3D-GANs. Generated images can be manipulated independently across shapes, expressions, abledos, illumination conditions as well as hairstyles and backgrounds.

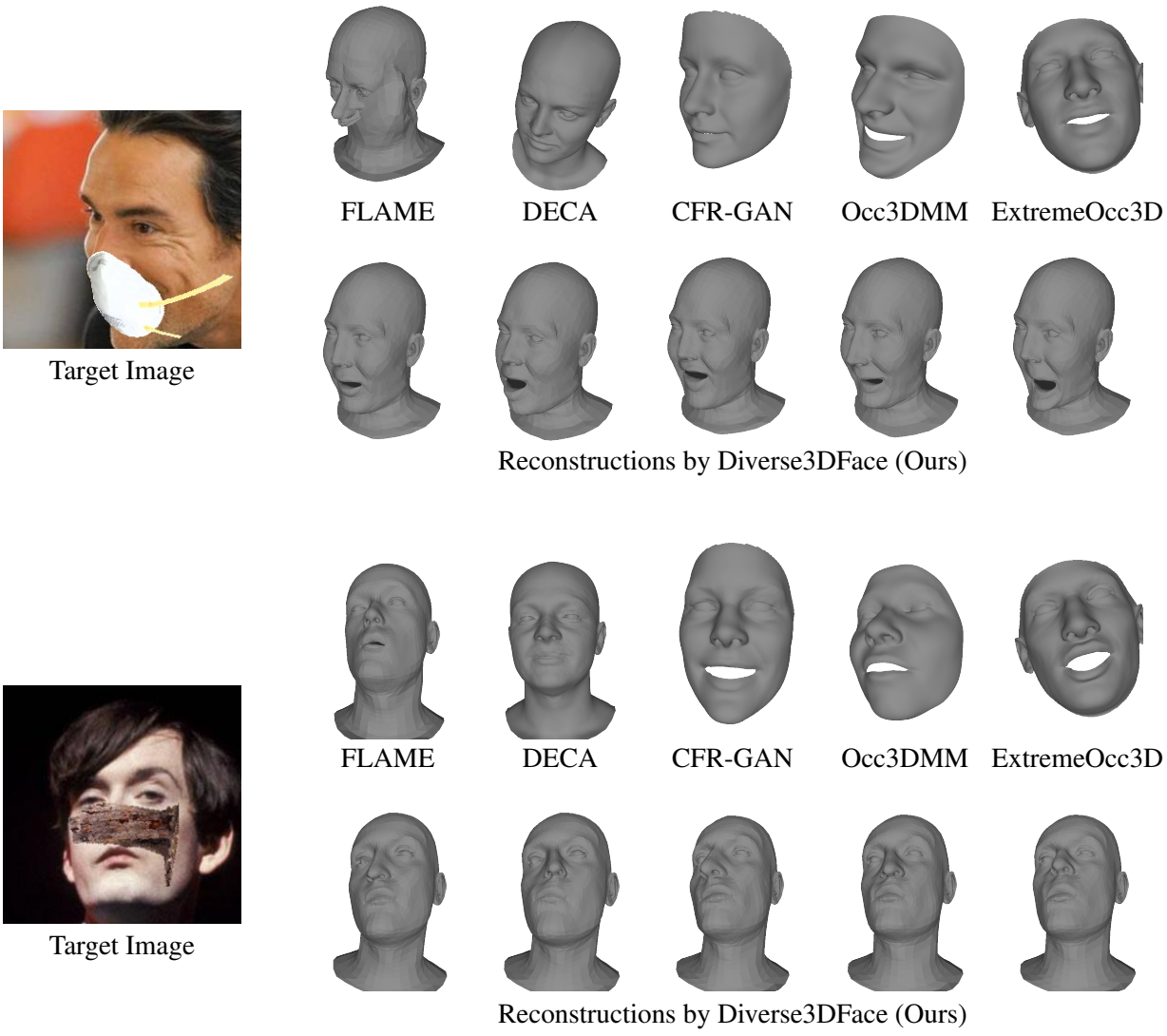


Figure 1.5 Diverse 3D reconstructions from a single occluded face image by Diverse3DFace *vs.* singular solution by the baselines including FLAME-Fitting [Li et al., 2017a], DECA [Feng et al., 2021], CFR-GAN [Ju et al., 2022], Occ3DMM [Egger et al., 2018], and ExtremeOcc3D [Tuán Trán et al., 2018].

based [Tran and Liu, 2019, Tran et al., 2019, Tuán Trán et al., 2018, Wu et al., 2020, Sengupta et al., 2018], rely on a global model to reconstruct a 3D model from a face image. This is not optimum in the presence of occlusion as the global model is either affected by occlusion, or it needs to be heavily regularized, which leads to sub-optimal 3D reconstruction (see Fig. 1.5). Further, these approaches generate a single solution even in the presence of occlusion. In contrast, we propose a global+local model that separates shape fitting on visible facial regions from those that are occluded, resulting in higher accuracy in the reconstruction of visible parts. We follow this by a diversity-oriented shape completion of the occluded parts, using a mesh-based VAE [Zhou et al., 2020b] and a diversity loss based on the concept of determinantal point processes (DPP) [Kulesza and Taskar, 2012]. Extensive experiments demonstrate that, on face images occluded by masks, glasses, and other random objects, our approach generates a distribution of 3D shapes having $\sim$50% higher diversity on the occluded regions compared to the baselines. Moreover, our closest sample to the ground truth has $\sim$40% lower MSE than the singular reconstructions by both occlusion-aware baselines [Egger et al., 2018, Tuán Trán et al., 2018], and non-occlusion aware baselines [Li et al., 2017a, Feng et al., 2021].

1.1 Contributions

Our specific contributions are the following:

- We explore and present ways to leverage explicit 3D face modeling in generative tasks such as face inpainting and controllable 3D face generation
- In the context of 3D-aware face inpainting, we propose 3DFaceFill [Dey and Boddeti, 2022a] which disentangles the partial face into its 3D components to aid in face completion, thereby generating geometrically and photometrically better completions than baselines.
- We present a method called CoLa-SDF which leverages 3D modeling for controlled generation and manipulation of high-fidelity 3D faces, from which photorealistic 2D images can be rendered in multiple views.

- We explore the problem of monocular 3D face reconstruction in the presence of occlusions by focussing on (i) robustness to occlusion and scene variations such as shape, pose, illumination, *etc*., and (ii) diversity of solutions rather than a single solution. To this end, we propose Diverse3DFace [Dey and Boddeti, 2022b] that employs an ensemble of global+local shape models that disentangle fitting on the visible regions from the occluded regions, followed by diversity-oriented completion of the occluded regions using the power of DPP [Kulesza and Taskar, 2012].

- We perform extensive quantitative and qualitative experiments to show the effectiveness of our proposed 3D-based approaches of face inpainting and controllable 3D face generation, and of our occlusion-aware 3D face reconstruction approach.

CHAPTER 2

BACKGROUND

In this chapter, we introduce the two main 3D modeling techniques we work with in this thesis: the 3D morphable models of faces, and implicit 3D models. We then present previous approaches that have dealt with similar tasks in face inpainting, controllable face generation, and 3D face reconstruction.

2.1 3D Morphable Models of Faces

3D morphable models (3DMMs) are a popular technique in computer vision and graphics that aim to model the variation in shape and texture of 3D objects, typically faces. A 3DMM is a statistical model that represents the shape and appearance of a 3D object as a linear combination of basis shapes and textures. These basis shapes and textures are learned from a set of training data, usually a large set of 3D scans of faces, which allows the model to capture the natural variation in shape and appearance of the object. The model can be used for a variety of tasks, such as face reconstruction, facial expression analysis, and face recognition. It can also be used for generating new faces that are statistically similar to the training data.

Some of the popular 3DMM models for faces are the Basel Face Model (BFM) [Paysan et al., 2009, Gerig et al., 2018] and the Faces Learned with an Articulated Model and Expressions (FLAME) model [Li et al., 2017a] (see Fig. 2.1). Specifically, FLAME [Li et al., 2017a] defines a 3D shape as:

$$S_{\text{w/pose}}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}) = \mathcal{W}\left(S(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\alpha}), \boldsymbol{\theta}, W\right), \tag{2.1}$$

where the parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}$ represent the shape, expression, and pose parameters, respectively; $\mathbf{J} \in \mathbb{R}^{\mathbf{3K}}$ represents the locations of K face joints around which $S(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta})$ is rotated, and finally smoothed by the blend weights W. The un-aligned shape $S(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta})$ is obtained by adding up the contributions of shape, expression and pose variations on top of a template shape $\bar{S}$:

$$S(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}) = \bar{S} + B_S(\boldsymbol{\alpha}; \mathcal{S}) + B_P(\boldsymbol{\theta}; \mathcal{P}) + B_E(\boldsymbol{\beta}; \mathcal{E}) \tag{2.2}$$

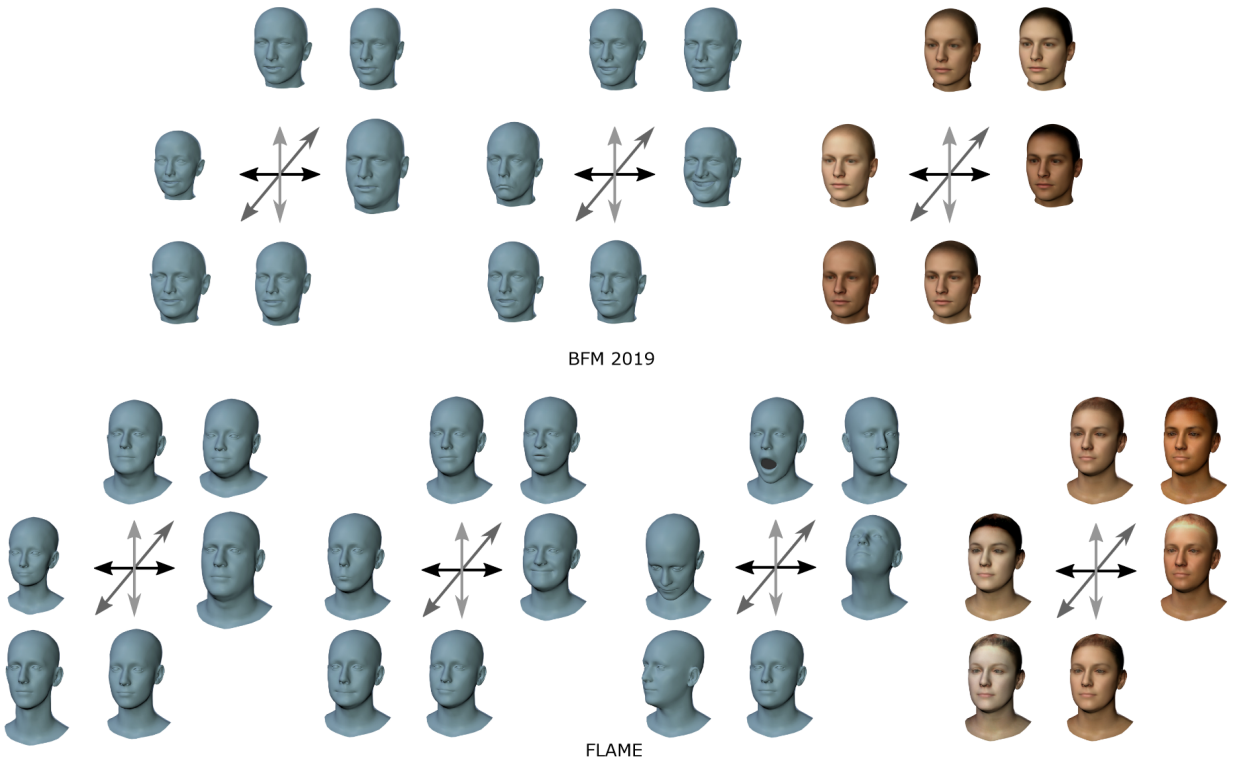


Figure 2.1 FLAME [Li et al., 2017a] and BFM [Gerig et al., 2018] 3DMMs. Image is sourced from [Egger et al., 2020].

The shape and expression variations are modeled by linear blendshapes $B_S(\boldsymbol{\alpha}; \mathcal{S}) = \mathcal{S}\boldsymbol{\alpha}$ and $B_E(\boldsymbol{\beta}; \mathcal{E}) = \mathcal{E}\boldsymbol{\beta}$, where $\mathcal{S} \in \mathbb{R}^{3N \times |\boldsymbol{\alpha}|}$ and $\mathcal{E} \in \mathbb{R}^{3N \times |\boldsymbol{\beta}|}$ are orthonormal shape and expression bases, respectively learned using PCA and N is the number of vertices. The pose blendshape function is defined as $B_P(\boldsymbol{\theta}; \mathcal{P}) = (R(\boldsymbol{\theta}) - R(\boldsymbol{\theta}*))\,\mathcal{P}$, where $R(\boldsymbol{\theta})$ comprises of rotation matrices around the K joints and $\mathcal{P} \in \mathbb{R}^{3N \times 9K}$ are the pose blendshapes describing the vertex offsets from the rest pose activated by R. Further, texture is modeled by linear combination of a set of texture basis $\mathcal{T}$: $B_T(\boldsymbol{\tau}; \mathcal{T}) = \mathcal{T}\boldsymbol{\tau}$.

3D face reconstruction using 3DMMs can be done from single view or multiview face images. We focus on single view 3D face reconstruction in this dissertation. For this, first a set of 2D facial landmarks are detected in the input image. These landmarks are used to normalize the pose and scale of the 2D face. Then, the parameters of the 3DMM are optimized to best fit the input image in terms of facial landmarks and appearance.

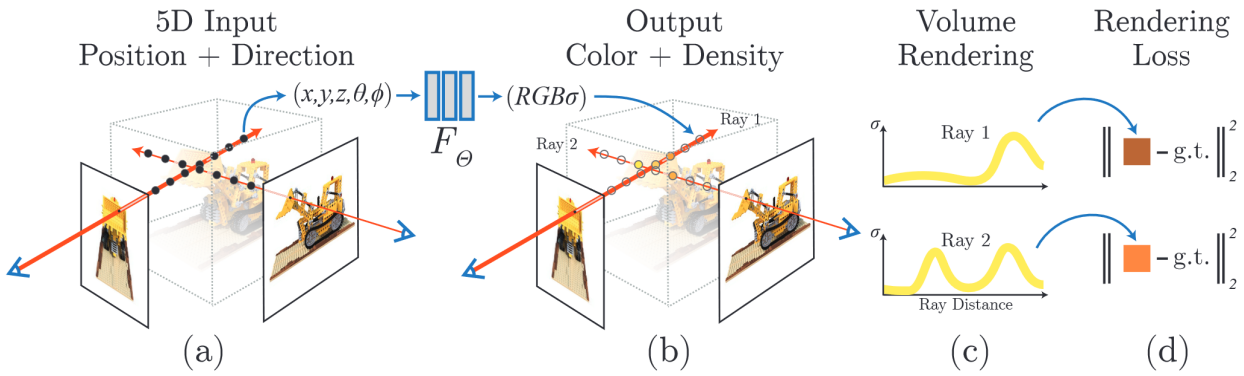


Figure 2.2 NeRF [Mildenhall et al., 2020] is a prominent implicit 3D model. The image synthesis process consists of (a) sampling 3D coordinates along camera-rays in the given viewing direction, (b) feeding these points and viewing directions to a neural network to obtain a set of color and density values, and (c) integrate and compose these rays into 2D images using volume rendering techniques. Image is sourced from the original paper.

2.2 Implicit 3D Models

Implicit 3D models such as NeRF[Mildenhall et al., 2020] refer to a type of 3D shape representation that defines the shape of an object as a continuous, implicit function. In other words, instead of representing a 3D object as a mesh of vertices and polygons or a point cloud, an implicit 3D model represents the object as a function that takes a 3D coordinate $\mathbf{x}$, and often a viewing direction $\mathbf{v}$, as inputs and outputs the radiance, and either the volume density or the signed distance value of that point. The implicit function is often modeled as a neural network. The volume density represents the opaqueness of the points, while the signed distance value indicates the distance between the input point and the surface of the object. If the signed distance is positive, the point is outside the object, and if it is negative, the point is inside the object. The surface of the object is defined as the set of points where the signed distance value is zero.

To render an image from a particular viewpoint, these approaches perform ray-marching to sample a set of 3D coordinates corresponding to each pixel in the image. This set of points, along with the corresponding viewpoint, is passed to a neural network to obtain the color and density values at these locations. Finally, classical volume rendering techniques are applied to render these points into a 2D image (refer Fig. 2.2).

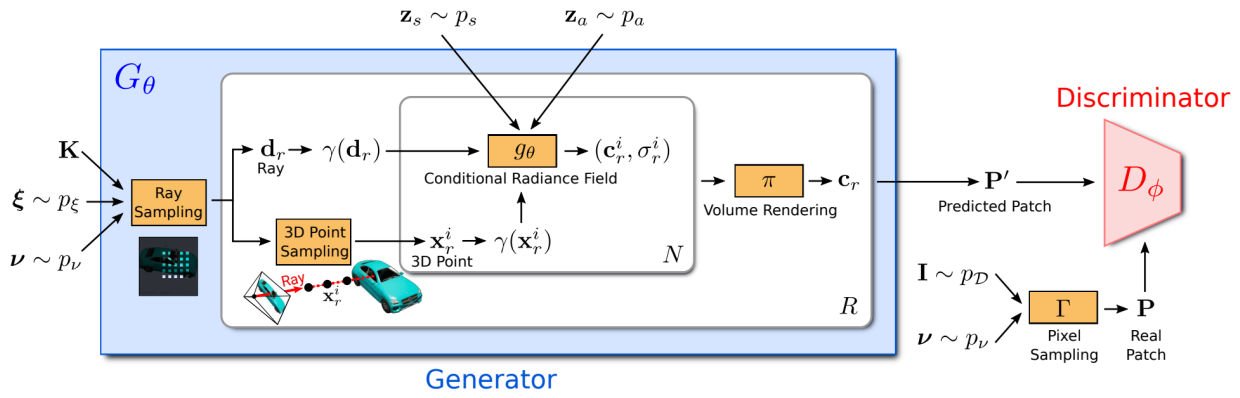


Figure 2.3 GRAF [Schwarz et al., 2020] is an implicit 3D-GAN that takes in a shape code and an appearance code, and samples a set of 3D coordinates conditioned on the camera parameters, and renders 2D images using volume rendering. It is trained in an adversarial manner.

2.3 Implicit 3D-GANs

Implicit 3D-GANs [Schwarz et al., 2020, Niemeyer and Geiger, 2021, Chan et al., 2021, Gu et al., 2021, Or-El et al., 2022] are a type of generative adversarial networks (GANs) that can be used to learn implicit 3D representations of objects. They combine the benefits of implicit 3D models with the power of GANs to generate new, realistic 3D shapes.

Unlike traditional GANs, which learn to generate images or 2D representations of objects, implicit 3D-GANs learn to generate 3D shapes as a continuous, implicit function. They take as input a latent vector, a set of 3D coordinates, and a viewing direction and output the color, and volume density or signed distance value at each 3D coordinate. From this, 2D images can be generated using the volume rendering techniques mentioned earlier [Mildenhall et al., 2020].

Implicit 3D-GANs are trained using a combination of adversarial and 3D regularization losses, which encourage the GAN to generate shapes that are both realistic and close to the true surface of the object. They can be used for a variety of tasks, including 3D shape generation, shape completion, and shape editing. The main advantage of implicit 3D-GANs is that they can generate new, realistic shapes that are not limited to the training data, with complex, intricate details and smooth surfaces.

2.4 Related Work

Due to their wide-ranging applications, both generative and 3D face modeling have seen a lot of research in the recent years, which have led to several advancements. They have also benefited vastly from advancements in related non-face specific approaches. We now review some of the related work in the areas of image inpainting, face inpainting, 3D face reconstruction, implicit 3D-GANs, editable implicit 3D models, and diversity promoting generating models.

2.4.1 Image Inpainting

Earlier image inpainting approaches[Bertalmio et al., 2000, Criminisi et al., 2004, Barnes et al., 2009, Hays and Efros, 2007] used diffusion or patch based methods to fill in the missing regions. This produced sharp results but often lacked semantic consistency. Recent techniques employ a CNN autoencoder along with a GAN loss to generate semantically consistent and realistic completions [Pathak et al., 2016, Yeh et al., 2017, Iizuka et al., 2017]. More recent methods focus on architectural enhancements to improve inpainting for variable and free form masks. These include a more refined discriminator in PatchGAN [Isola et al., 2017], contextual attention in DeepFillv2 [Yu et al., 2018] and gated convolutions [Liu et al., 2018, Yu et al., 2019]. In contrast, our work in Chapter 3 adopts vanilla CNN architectures and instead relies on a more accurate 3D face analysis-by-synthesis technique.

2.4.2 Face Inpainting

Face inpainting is a more challenging variant of image inpainting because of the complexity and diversity of faces. To address this, many approaches impose additional geometric and photometric priors in the form of face related losses [Song et al., 2019a, Li et al., 2017b, Chen et al., 2017b, Zhang et al., 2017, Li et al., 2020b, Yuan and Park, 2019]. A recent approach called DSA [Zhou et al., 2020a] uses oracle-learned attention maps and component-wise discriminators to generate high-fidelity completions. While it often generates photorealistic completions in well-lit frontal faces, it still relies on implicitly learned priors which are insufficient to enforce correct geometry in challenging poses and illuminations. All these approaches rely on novel architectural advances and loss functions while our method 3DFaceFill focuses on more explicit and precise modeling of

the image-formation process.

Concurrently, [Deng et al., 2018] completed self-occluded UV texture to synthesize new face views. This assumes that the full face image and at least half of the UV texture is always visible. In contrast, our 3DFaceFill goes beyond self-occlusion and instead, performs 3D factorization on the masked face and completes its *albedo* for *masked face completion*. Furthermore, since texture is not always symmetric due to illumination variations, [Deng et al., 2018] needs synthetically completed texture maps for training; whereas 3DFaceFill performs completion on albedo which is further disentangled from both geometry as well as illumination allowing us to effectively enforce symmetry prior, without needing synthetically completed UV-maps for training, as it bears out in our experiments. A few recent works have also attempted to leverage symmetry for face completion [Zhang et al., 2018a, Li et al., 2020a]. However, these approaches employ complex symmetry registration operations, which require huge computational resources; moreover these operations are often susceptible to large geometric variations.

2.4.3 Linear and Nonlinear 3DMMs

Blanz and Vetter [Blanz and Vetter, 1999] proposed the first statistical 3DMM of human faces. Since then, such models have grown to include complex pose, expression, and texture modalities in faces [Paysan et al., 2009, Gerig et al., 2018]. FLAME, proposed by [Li et al., 2017a], models the full human head and allows non-linear control over joint poses to generate articulated expressive head instances. While relatively simple and effective, these linear models often lack expressivity and detail. Over the past several years, many approaches began adopting neural networks to model higher-order complexities in the shape and texture spaces [Tewari et al., 2017, Sengupta et al., 2018, Shu et al., 2017, Tran and Liu, 2019, Tran et al., 2019, Tuan Tran et al., 2017, Ramon et al., 2021, Kim et al., 2018, Sanyal et al., 2019]. [Wu et al., 2020] leveraged facial symmetry and illumination to learn a 3D model of faces from in-the-wild images in an unsupervised way. [Medin et al., 2022] trained a nonlinear 3DMM, called MOST-GAN, to integrate the expressiveness of style-based GANs with the physical disentanglement of 3DMMs, along with a 2D hair manipulation network. Some approaches took a coarse-to-fine approach to add details to 3D re-

constructions. DECA [Feng et al., 2021] adds a pose and expression conditioned displacement map on top of a coarse shape to make the 3D reconstructions animatable. [Grassal et al., 2022] employed a coarse mesh refinement approach to learn subject-specific head avatars that model the entire head including hair.

Motivated by the advances in graph neural networks [Kipf and Welling, 2016, Veličković et al., 2017, Defferrard et al., 2016, Morris et al., 2019], some recent approaches adopted graph convolutions to directly learn nonlinear representations on a mesh surface, while preserving the mesh topology [Ranjan et al., 2018, Bouritsas et al., 2019, Zhou et al., 2020b]. A few methods took a hybrid approach of fitting a non-linear neural network model to the target image to generate detailed 3D reconstructions [Gecer et al., 2019, Yenamandra et al., 2021].

However, compared to implicit 3D-GANs, these models do not generate as high quality and intricately detailed 3D faces. Further, they have limited modeling of hair and teeth since these facial regions lack pointwise correspondence across subjects and are not part of the underlying 3DMM models. Also, these approaches are not designed explicitly to handle occlusions. Hence, when used for 3D reconstruction, these approaches often produce artifacts and lead to poor shape and pose estimation in the presence of facial occlusions.

2.4.4 Occlusion-Robust 3D Face Reconstruction

To improve occlusion robustness during 3D face reconstruction, a few approaches are explicitly designed to handle occlusions [Tuán Trán et al., 2018, Egger et al., 2018, Ju et al., 2022, Li et al., 2023]. [Tuán Trán et al., 2018] trained a neural network to regress a robust foundation shape from a masked face image, over which a detailed bump map is added later. [Egger et al., 2018] employed an EM-like approach to simultaneously optimize an occlusion mask and the model parameters for a target occluded image. [Li et al., 2023] adopted this strategy of 3D reconstruction aiding in occlusion segmentation and vice versa, to simultaneously train a face encoder and an outlier segmentation network. However, these approaches rely on a global model to account for the entire face, including the occluded parts, which is sub-optimal as the lack of information from such parts needs to be countered using strong regularization. Moreover, they are limited to reconstructing a

singular 3D solution without considering the plurality of solutions that can explain the occluded regions. In contrast, our proposed Diverse3DFace addresses the dual problems of robustness and lack of uniqueness through a multistage approach that disentangles fitting on the visible regions from diversity modeling on the occluded ones.

2.4.5 Diversity Promoting Generative Models

Diversity promoting algorithms have been employed in several areas in computer vision where a distribution of outcomes is more desirable than a singular solution. Conditioning [Isola et al., 2017, Yang et al., 2019] and regularization [Zhu et al., 2017, Ghosh et al., 2018, Suzuki et al., 2016, Che et al., 2016, Srivastava et al., 2017] based techniques have been proposed to overcome mode-collapse and promote diversity in GANs [Goodfellow et al., 2014]. As ill-posed problems, diversity promoting algorithms are particularly useful for image inpainting and image super-resolution. [Zheng et al., 2019b] introduced the notion of diversity of solutions in image inpainting. They proposed a dual-pipeline C-VAE [Sohn et al., 2015] that maintains ground-truth fidelity in one path while allowing diversity on the other. [Bahat and Michaeli, 2020] generated diverse super-resolution explanations by only enforcing consistency in the low-resolution space. As one of the most seminal works in this field, [Kulesza and Taskar, 2012] introduced the framework of Determinantal Point Processes (DPPs) to model diversity in machine learning tasks such as inference, sampling, marginalization, *etc*. [Yuan and Kitani, 2019, Yuan and Kitani, 2020] adopted DPP to sample multi-modal latent vectors for diverse human trajectory forecasting. [Elfeki et al., 2019] devised a DPP-based objective to train GANs and VAEs to emulate the diversity in real data. In Chapter 5, we adopt the idea of DPPs to generate diverse 3D reconstructions for an occluded face by discovering latent space representations that maximize plausible diversity on the occluded regions while remaining faithful to the visible parts.

2.4.6 Implicit Neural Representations and 3D-GANs

Instead of explicitly representing objects and scenes as meshes, voxel grids or point clouds, implicit 3D models represent them through the parameters of a neural network. [Mildenhall et al., 2020] proposed the first method of neural radiance fields (NeRFs), in which the density and radi-

ance of 3D points are queried through the network and rendered to an image using volume rendering. NeuS [Wang et al., 2021] adopted signed distance fields instead of density fields to represent the object surfaces. While these models are fitted to a given scene and are not generative in nature, several later approaches adopted neural rendering to learn implicit 3D-GANs [Schwarz et al., 2020, Niemeyer and Geiger, 2021]. pi-GAN [Chan et al., 2021] proposed a novel architecture based upon periodic activation function [Sitzmann et al., 2020] and feature-wise linear modulation (FiLM) to improve view consistency and generation quality of implicit 3D-GANs. While these methods were successful in generating 3D scenes that can be rendered to view-consistent images, high computational cost prevented them from generating high-resolution images. EG3D [Chan et al., 2022] introduced a tri-planar framework and showed that it improves computational efficiency and multi-view consistency of generated images. More recent methods like StyleNeRF [Gu et al., 2021] and StyleSDF [Or-El et al., 2022] have adopted a hybrid approach of combining a low-resolution volume renderer with a CNN-based super-resolution network. Although these approaches enable direct manipulation of the 3D viewpoint, they otherwise lack any explicit control over the generated objects.

2.4.7 Editable Implicit 3D Models

There have been several attempts to enable editing of implicit 3D-GANs. BANMo [Yang et al., 2022] learned a neural blend skinning model to transform 3D points between the camera space and a learned canonical space, enabling large deformations. NeRF-Editing [Yuan et al., 2022] utilized ray-bending to edit the underlying static NeRF. HeadNeRF [Hong et al., 2021] disentangled the latent space of an implicit 3D-GAN for faces by training on data containing multiple images for each subject with the same labeled variations in expression and illumination. StyleRig [Tewari et al., 2020b] and PIE [Tewari et al., 2020a] embed portrait images into the latent space of the pretrained StyleGAN model [Karras et al., 2020, Karras et al., 2021] for editing. CLIP-NeRF [Wang et al., 2022] performs text- or exemplar-based editing of low-resolution objects. Disentangled3D [Tewari et al., 2022] and FENeRF [Sun et al., 2022] train separate shape deformation and appearance networks, but they do not disentangle illumination and only generate low-resolution images. RigNeRF

[Athar et al., 2022] enables editing of portraits by learning a deformation NeRF with respect to a canonical space modeled by a 3DMM, but it is subject-specific and does not allow for generating new identities. However, compared to our method in Chapter 4, these methods lack explicit semantic control over specific aspects of the face, and often lack photorealism.

CHAPTER 3

3DFACEFILL: AN ANALYSIS-BY-SYNTHESIS APPROACH TO FACE COMPLETION

Dey, R. and Boddeti, V. N. 3DFaceFill: An Analysis-by-Synthesis Approach to Face Completion. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1586–1595, 2022.

In this chapter, we explore the applicability of 3D face modeling for face inpainting (also known as completion). End-to-end image completion methods i.e., models that generate 2D completions directly from 2D masked images, have witnessed remarkable progress in recent years. These approaches rely primarily on architectural advances in neural network designs to implicitly account for photometric and geometric variations in image appearance. And even those that explicitly include scene geometry in their formulation do so largely in 2D. Consequently, object-based image completions from such methods often suffer from poor photorealism, especially under large variations in pose, shape, illumination of objects in the image and the inpainting mask. For example, in the context of faces, Fig. 1.3 shows face images having extreme poses (1.3.A), illumination variations across the face (1.3.C) and diverse appearances and shapes. Current state-of-the-art methods such as DeepFillv2 [Yu et al., 2019] and PICNet [Zheng et al., 2019a], both of which operate end-to-end on 2D image representations, often fail in preserving facial symmetry and the variations of the aforementioned factors (pose, illumination, texture, shape) while inpainting.

Several attempts have been made to customize generic image inpainting solutions for structured objects such as faces. General image inpainting approaches typically employ a CNN autoencoder as the inpainter and train it using a combination of photometric and adversarial losses [Pathak et al., 2016, Iizuka et al., 2017, Yu et al., 2018, Zheng et al., 2019a]. Face specific completion methods [Li et al., 2017b, Song et al., 2019a] employ additional losses such as landmark loss, perceptual loss and face parsing loss. However, these approaches still do not account for all factors in the image formation process like illumination and pose variations and as such fail to effectively

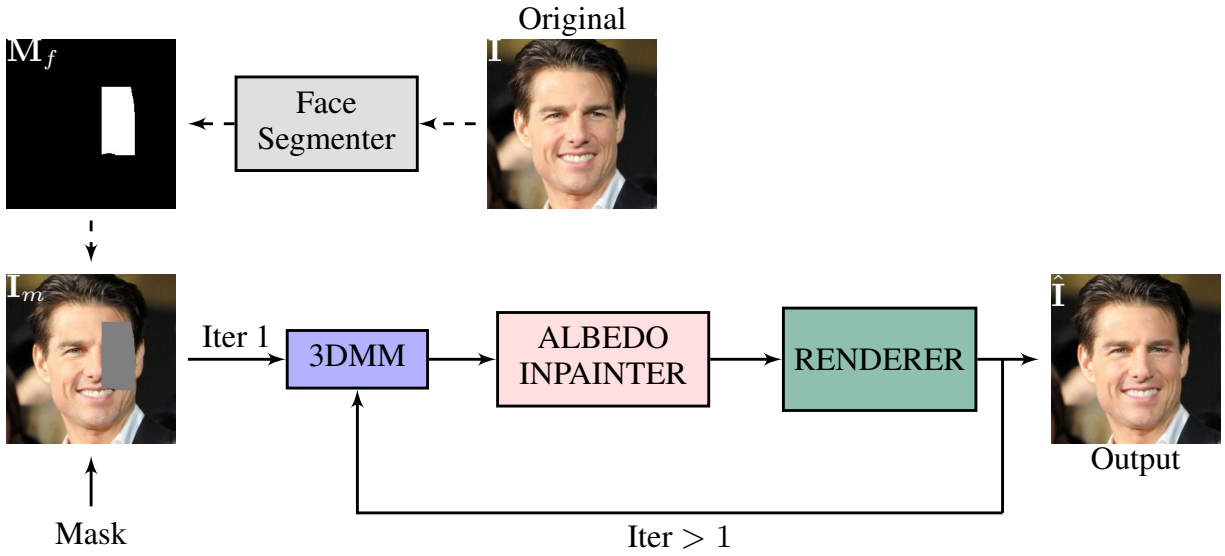


Figure 3.1 Overview: 3DFaceFill is an iterative inpainting approach where the masked face is disentangled into its 3D shape, pose, illumination and partial albedo by the 3DMM module, following which the partial albedo is inpainted and finally the completed image is rendered. During inference (only), the completed image is fed back through the whole pipeline in subsequent iterations, while using the initial mask for albedo inpainting. During training, a pre-trained model segments the image into face, hair and background for constraining the mask to lie only on the face. This segmentation is optionally used during inference if necessary.

impose geometric priors such as facial symmetry. Moreover, the implicit enforcement of geometric priors is still done in 2D as opposed to in 3D. This is a significant limitation as faces are inherently symmetric 3D objects and their projections on 2D images are often affected by the aforementioned factors of pose, illumination, shape *etc*.

In contrast to the foregoing, our approach advocates for an analysis-by-synthesis approach for face completion that explicitly accounts for the 3D structure of faces i.e., shape and albedo, and image formation factors i.e., pose and illumination. The key insight of our solution is that performing face completion on the UV representation, as opposed to the 2D pixel representation, allows us to effectively leverage the power of correspondence and ultimately lead to geometrically and photometrically accurate face completion (see Fig.1.3). Our approach (see Fig. 3.1), dubbed 3DFaceFill, comprises of three components that are iteratively executed. First, the masked face image is disentangled into its constituent geometric and photometric factors. Second, an autoencoder performs inpainting on the UV representation of facial albedo. Lastly, the completed face is

re-synthesized by a differentiable renderer. Our specific contributions are:

– We propose 3DFaceFill, a simple yet very effective face completion model that explicitly disentangles photometric and geometric factors and perform inpainting in the UV representation of facial albedo while preserving the associated facial shape, pose and illumination.

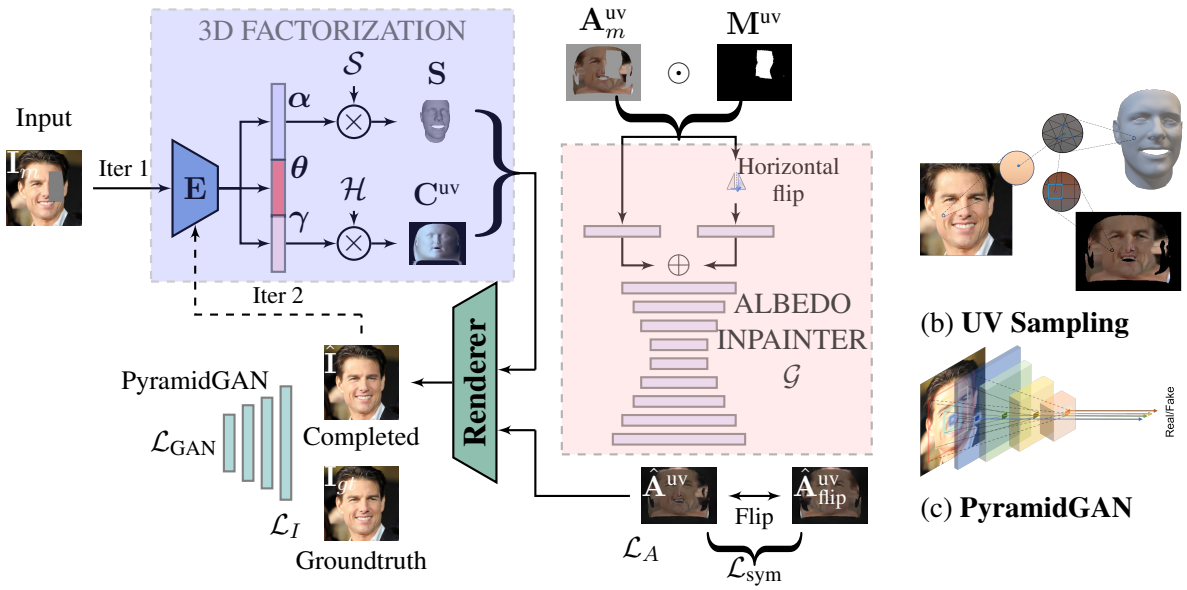
– We propose a 3D symmetry-aware network architecture and a symmetry loss for the inpainter to propagate albedo features from the visible to symmetric masked regions of the UV representation. Enforcing the symmetry prior in 3D, as opposed to 2D, allows 3DFaceFill to more effectively leverage and preserve facial *symmetry* while inpainting.

– Given our trained model, we propose a simple refinement process at inference by *iteratively* reprocessing the face completion through the model. This process enables us to address the "chicken-and-egg" problem of simultaneously inferring both the photometric and geometric factors and completion of the face from a masked image. The procedure is especially effective for heavily masked faces, improving the PSNR by up to 1dB.

– Extensive benchmarking on several datasets and unconstrained in-the-wild images results in 3DFaceFill producing photorealistic and geometrically consistent face completions over a range of masks and real occlusions, especially in terms of pose, lighting, and attributes such as eye-gaze and shape of nose along with a quantitative improvement of upto 4dB PSNR and 25% in LPIPS[Zhang et al., 2018b].

3.1 Approach

In this section, we first present an overview of our proposed 3D face completion approach (dubbed 3DFaceFill) followed by the details of each component. As shown in Fig. 3.1, 3DFaceFill has three components: a 3DMM encoder, an albedo completion module and a renderer. Given a masked face, 3DFaceFill first resolves it into its constituent 3D shape, pose and illumination using the 3DMM encoder (Fig. 3.2). Then, we obtain the partial facial texture in the UV-domain by re-projecting the mesh onto the input image (Fig. 3.2b). We further remove the shading component to obtain an illumination-invariant partial albedo. The inpainter completes the partial albedo using symmetric and learned priors. Finally, the renderer combines the inpainted albedo with the esti-



(a) **Architecture**

Figure 3.2 (a) Architecture: Given a masked face $\mathbf{I}_m$, the 3DMM encoder extracts its shape parameters $\boldsymbol{\alpha}$, pose $\boldsymbol{\theta}$ and illumination parameters $\boldsymbol{\gamma}$, from which we obtain the full shape $\mathbf{S} = \mathcal{S}\boldsymbol{\alpha}$, and shade represented in UV $\mathbf{C}^{\text{uv}} = \mathcal{H}\boldsymbol{\gamma}$ by linear combination with the corresponding orthonormal shape and spherical harmonics bases $\mathcal{S}$ and $\mathcal{H}$, respectively. Then, we obtain a partial albedo $\mathbf{A}^{\text{uv}}_m$ as shown on the right in **(b)** by first, re-projecting the 3D mesh onto the masked image to obtain the UV-texture $\mathbf{T}^{\text{uv}}_m$, and then, removing the shade from it $\mathbf{A}^{\text{uv}}_m = \mathbf{T}^{\text{uv}}_m \oslash \mathbf{C}^{\text{uv}}$. Finally, the albedo inpainter $\mathcal{G}$ completes the partial albedo as $\hat{\mathbf{A}}^{\text{uv}}$, conditioned on the UV-mask $\mathbf{M}^{\text{uv}}$. We then combine the completed albedo with the estimated shape, pose and shade to obtain the completed image $\hat{\mathbf{I}}$. To generate photorealistic completion, the completed and groundtruth images are evaluated by the proposed (c) PyramidGAN discriminator. (b) UV Sampling: 3D mesh is projected onto the face image to obtain per vertex RGB values $\mathbf{T}_v(\mathbf{v})$. We map the per-vertex texture map to a UV texture map $\mathbf{T}^{\text{uv}}$ using a pre-defined mapping.

mated 3D factors to obtain the completed face. As a natural extension of the proposed approach, we use 3D factorization and completion in a complimentary way to further improve completion iteratively.

3.1.1 3D Factorization

Existing face image completion approaches directly operate on 2D, which makes it non-trivial to enforce strong 3D geometric and photometric priors. This leads to poor face completion in challenging conditions of poses, geometry, lighting, *etc*. This motivates us to adopt explicit 3D factorization of face images to disentangle the appearance and geometric components, to enable

robust completion.

Essentially, the 3D factorization module is an inverse renderer $\Phi : \mathbf{I} \rightarrow (\mathbf{S}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \mathbf{A})$ that resolves a 2D face $\mathbf{I}$ into its constituent 3D shape $\mathbf{S} \in \mathbb{R}^{N\times 3}$, 3D pose $\boldsymbol{\theta} = (s, R, \mathbf{t})$, illumination $\boldsymbol{\gamma}$ and albedo $\mathbf{A}$. Various 3DMM approaches like [Blanz and Vetter, 1999, Egger et al., 2018, Gecer et al., 2019] can be a natural fit for this. However, being fitting based approaches, they are not real time, leaving learning based 3D reconstruction approaches [Tewari et al., 2017, Sengupta et al., 2018, Shu et al., 2017, Tuán Trán et al., 2018, Tran and Liu, 2019, Tran et al., 2019, Wu et al., 2020] as the obvious choices. While any of these approaches can potentially be used in our approach, for the purpose of this work, we adopt a simplified version of the nonlinear 3DMM presented in *et al*. [Tran and Liu, 2019].

The 3D factorizaiton module consists of a 3DMM encoder $\mathcal{E}$ and an albedo decoder $\mathcal{G}_A$ (used only during training). The encoder $\mathcal{E}$ first resolves the image $\mathbf{I}$ in to its shape $\boldsymbol{\alpha}$, albedo $\boldsymbol{\tau}$ and illumination $\boldsymbol{\gamma}$ parameters, and its 3D pose $\boldsymbol{\theta} = (s, R, \mathbf{t})$. Using the shape coefficients, we obtain the full 3D shape $\mathbf{S}$ by linear combination with the Basel Face Model's (BFM) [Paysan et al., 2009] orthonormal shape bases $\mathcal{S}$: $\mathbf{S} = \mathcal{S}\boldsymbol{\alpha}$, where $\mathcal{S} \in \mathbb{R}^{3N\times|\boldsymbol{\alpha}|}$ and N is the number of vertices. Similarly, we combine the illumination coefficients linearly with the spherical harmonics (SH) bases $\mathcal{H}$ [Ramamoorthi and Hanrahan, 2001] to obtain the surface shading in the UV-domain: $\mathbf{C}^{\text{uv}} = \mathcal{H}\boldsymbol{\gamma}$, where $\mathcal{H} \in \mathbb{R}^{H\times W\times 3\times 9}$ (9 bases per color channel), and H and W are the height and width of the UV-representation, respectively. The decoder $\mathcal{G}_A$ maps the albedo coefficients into the full UV-albedo $\mathcal{G}_A : \boldsymbol{\tau} \rightarrow \mathbf{A}^{\text{uv}}$, which is then multiplied with the shade to obtain the texture $\mathbf{T}^{\text{uv}} = \mathbf{A}^{\text{uv}} \odot \mathbf{C}^{\text{uv}}$. A differentiable renderer $\mathcal{R}$ [Tran and Liu, 2019] then re-projects the estimated 3D factors into image $\mathbf{I}_{ren}$ using the Z-buffer technique:

$$\mathbf{I}_{ren} = \mathcal{R}\left(\mathbf{S}, \mathbf{T}^{\text{uv}}, \boldsymbol{\theta}\right) \tag{3.1}$$

We train the module using masked images for robustness to partial inputs. For further details, refer the appendix.

3.1.2 Albedo Completion Module

Architecturally, our albedo completion module is similar to other adversarially trained image-completion autoencoders [Pathak et al., 2016, Li et al., 2017b, Yu et al., 2018]. However, ours has the unique advantage of being solely focused on recovering the missing albedo, which has been disentangled from other variations in shape, pose and illumination through 3D factorization and is largely symmetric in its UV-representation. UVGAN [Deng et al., 2018] performs a similar completion of self-occluded UV-texture extracted from fully-visible face images. However, because of the entangled illumination, they don't use symmetry and need a synthetically completed texture map for supervision, whereas we use symmetry as self-supervision.

To this end, we discard the soft albedo obtained from the 3DMM albedo decoder and instead obtain the more realistic partial albedo from the input image in the UV space. This is done in two steps: first, we reproject the obtained 3D mesh onto the face image and use bilinear interpolation to sample the per-vertex texture (see Fig. 3.2b):

$$\mathbf{T}^{\mathrm{v}}_m(x, y, z) = \sum_{\substack{p\in\{\lfloor x\rfloor,\lceil x\rceil\}\\ q\in\{\lfloor y\rfloor,\lceil y\rceil\}}} \mathbf{I}^{p,q}_m(1-|x-p|)(1-|y-q|) \tag{3.2}$$

Then, we map the sampled partial texture $\mathbf{T}^{\mathrm{v}}_m$ onto the UV space using barycentric interpolation on the predefined mesh-to-uv mappings $\mathbf{T}^{\mathrm{v}}_m(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) \rightarrow \mathbf{T}^{\mathrm{uv}}_m(u, v)$. From the texture, we obtain the partial albedo by simply removing the estimated shade: $\mathbf{A}^{\mathrm{uv}}_m = \mathbf{T}^{\mathrm{uv}}_m \oslash \mathbf{C}^{\mathrm{uv}}$, where $\oslash$ is the element-wise division operation. We perform similar operations to unwarp the mask $\mathbf{M}$ on-to the UV-space as $\mathbf{M}^{\mathrm{uv}}$.

We use a U-Net [Ronneberger et al., 2015] based autoencoder $\mathcal{G}$ to complete the partial albedo conditioned on the input mask, $\mathcal{G} : (\mathbf{A}^{\mathrm{uv}}_m, \mathbf{M}^{\mathrm{uv}}) \rightarrow (\hat{\mathbf{A}}^{\mathrm{uv}}, \sigma^{\mathrm{uv}})$, where $\hat{\mathbf{A}}^{\mathrm{uv}}$ is the completed albedo and σ^{uv} is the uncertainty of completion. In order to leverage the bilateral symmetry of the UV facial albedo as an attention map, we modify the U-Net architecture (henceforth referred to as Sym-UNet). This is specially helpful since we do not have access to the full groundtruth albedo maps for training. To do so, we split the first convolution layer $f_{1:2c}$ into two parts: $f_{1,1:c}$ and $f_{2,c+1:2c}$ with equal number of output channels c (see Fig. 3.2). The first filter operates on the

input albedo as such to obtain the response $\mathbf{h}_1 = f_1(\mathbf{A}_m^{\text{uv}})$. The second, instead, operates on the horizontally flipped albedo $\mathbf{h}_2 = f_2(\text{hflip}(\mathbf{A}_m^{\text{uv}}))$. We then concatenate the responses $\mathbf{h}_1$ and $\mathbf{h}_2$ from these two filters and pass it through the rest of the network. During training, the first filter learns to extract features from the visible parts of the albedo while the second filter learns to extract features corresponding to the symmetrically opposite visible parts to apply on the occluded regions (see Fig. 3.15).

A naive approach of doing so, however, results in artifacts from the symmetrical counterparts to appear on the visible regions, making the network convergence difficult. Instead, we use gated convolutions [Yu et al., 2019] as shown in Fig. 3.15 (in all but the final layer), to ensure that such symmetric features are only transferred to the masked regions and do not create artifacts on the visible regions. We use group normalization[Wu and He, 2018] and ELU activation[Clevert et al., 2015] for all the feature layers and the final output is simply clipped between -1 and 1. We then render the completed albedo $\hat{\mathbf{A}}^{\text{uv}}$, along with the estimated shape, pose and illumination to obtain a completed image $\hat{\mathbf{I}}$ using eqn. 3.1. Finally, we simply blend the input and completed images to obtain the output image: $\mathbf{I}_{\text{out}} = \mathbf{I}_m \odot (1 - \mathbf{M}) + \hat{\mathbf{I}} \odot \mathbf{M}$.

PyramidGAN Discriminator: To generate sharp and semantically realistic completions, we use a multi-scale PatchGAN discriminator [Wang et al., 2018, Shocher et al., 2019], which we refer to as the *PyramidGAN*. The PyramidGAN evaluates the final output $\mathbf{I}_{\text{out}}$ at multiple locations and scales ranging from coarse and global to fine and local (refer to Fig. 3.2c). Features from each l-th downsampling layer of the PyramidGAN $\mathcal{D}_l$ are used to evaluate an average hinge loss [Yu et al., 2019, Juefei-Xu et al., 2018] for that layer. We then compute the average loss across all the layers as the total loss, thus giving equal weightage to each scale:

$$\begin{aligned} \mathcal{L}_{\mathcal{G}} =& - \mathbb{E}_{p(z)} \left[\mathbb{E}_{l \in L} \left[\mathcal{D}_l(\mathcal{G}(z)) \right] \right] \\ \mathcal{L}_{\mathcal{D}} =& \mathbb{E}_x \left[\mathbb{E}_{l \in L} [\mathbf{1} - \mathcal{D}_l(x)]_+ \right] + \mathbb{E}_{p(z)} \left[\mathbb{E}_{l \in L} [\mathbf{1} + \mathcal{D}_l(\mathcal{G}(z)]_+ \right], \end{aligned} \tag{3.3}$$

Training Losses: We train the albedo completion module with the following total loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_A + \lambda_2 \mathcal{L}_I + \lambda_3 \mathcal{L}_{\text{sym}} + \lambda_4 \mathcal{L}_{\text{GAN}} + \lambda_5 \mathcal{L}_{\text{gp}}, \tag{3.4}$$

where $\mathcal{L}_A = \mathcal{L}_\sigma(||\hat{\mathbf{A}}^{\text{uv}} - \hat{\mathbf{A}}^{\text{uv}}_{\text{gt}}||_1, \sigma^{\text{uv}})$ and $\mathcal{L}_I = \mathcal{L}_\sigma(||\hat{\mathbf{I}} - \mathbf{I}_{gt}||_1, \sigma)$ are the pixel losses for the albedo and the image, respectively, $\mathcal{L}_{\text{sym}}$ is the symmetry loss, $\mathcal{L}_{\text{GAN}}$ is the GAN loss given in eqn. 3.3 and $\mathcal{L}_{\text{gp}}$ is the WGAN-GP loss as described in [Gulrajani et al., 2017]. The albedo symmetry loss is carefully applied on the masked regions whose symmetric counterparts are visible, to supplement as supervised attention:

$$\mathcal{L}_{\text{sym}} = \mathcal{L}_\sigma \left((\mathbf{1} - \mathbf{M}^{\text{uv}}) \mathbf{M}^{\text{uv}}_{\text{flip}} \odot ||\hat{\mathbf{A}}^{\text{uv}} - \hat{\mathbf{A}}^{\text{uv}}_{\text{flip}}||_1, \sigma^{\text{uv}} \right) \tag{3.5}$$

Here, $\mathcal{L}_\sigma(\mathbf{x}, \sigma)$ is the aleatoric uncertainty loss[Kendall and Gal, 2017], given by:

$$\mathcal{L}_\sigma(\mathbf{x}, \sigma) = \frac{1}{\dim(\mathbf{x})} \sum_i \frac{1}{2} x_i e^{-\sigma_i} + \frac{\sigma_i}{2}. \tag{3.6}$$

The loss coefficients are set to have similar magnitude for all the loss components. In our approach, the goal is to show the efficacy of explicit 3D consideration on the geometric and photometric accuracy of face completion. So, *we withhold from using attention or face specific losses, or refiner modules that many other approaches have used* [Li et al., 2017b, Yu et al., 2018, Yu et al., 2019, Zheng et al., 2019a, Zhou et al., 2020a, Medin et al., 2022] and leave them as future add-ons.

Iterative Refinement: 3D factorization is an important first step of our proposed approach, which itself leads to robust face completion in cases where 2D based methods fail. To make the 3D factorization itself robust to partial images, we train the 3DMM encoder on face images with randomly sized and randomly located masks. However, there is scope to further improve upon this and leverage the full power of our proposed two-step approach. To do this, we adopt a simple iterative refinement technique where face completion leads to improved 3D factorization and vice versa, as shown in Fig. 3.1. During inference, the masked face is used to distill the 3D factors in the first iteration; while in the next iteration, the completed face itself forms the input for 3D analysis. This leads to iteratively refined 3D analysis (*specially the 3D pose*) as well as face completion. Though one can repeat the iterative step many times, we experimentally found that two such iterations are usually sufficient.

3.2 Experimental Evaluation

Datasets: We evaluate the proposed 3DFaceFill on the CelebA [Liu et al., 2015] and CelebA-HQ [Lee et al., 2020] datasets. We use 80% split for training and 20% for evaluation. Further, to evaluate the robustness and generalization performance, we do a cross-dataset evaluation on the pose and illumination varying images from the MultiPIE [Gross et al., 2010] dataset and $\sim$50 in-the-wild face images downloaded from the internet[1].

Implementation Details: We train both the 3D factorization and the completion modules independently using the Adam optimizer with a learning rate of 10^{-4}. We first train the 3DMM module on the 300W-3D [Zhu et al., 2016] and the CelebA [Liu et al., 2015] datasets. Once the 3DMM encoder is trained, we freeze it and use it to train the completion module on the CelebA [Liu et al., 2015] dataset for 30k iterations. We generate random rectangular masks of varying sizes and locations, and constrain them to lie in the segmented face region (Fig. 3.1). Please see the appendix Sec. A.3 for further details on implementation and computational analysis.

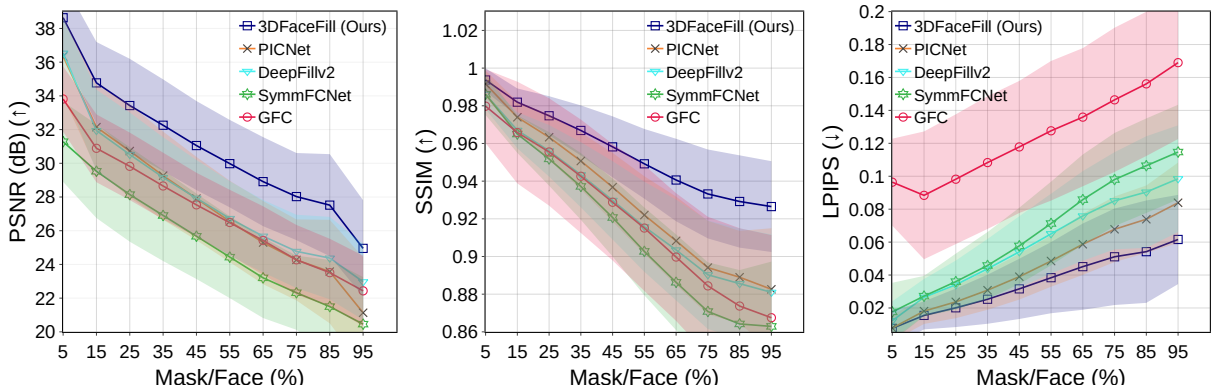
Baselines: To evaluate the efficacy of 3DFaceFill, we perform qualitative and quantitative comparison against baselines such as GFC [Li et al., 2017b], SymmFCNet [Li et al., 2020a], DeepFillv2 [Yu et al., 2019, Yu et al., 2018] and PICNet[2] [Zheng et al., 2019a]. We use the publicly available pretrained face models for DeepFillv2 [Yu et al., 2019], PICNet [Zheng et al., 2019a] and SymmFCNet [Li et al., 2020a]. For GFC [Li et al., 2017b], the pretrained model was not trained on the same crop and alignment as ours, so we train it from scratch using their source code. Due to the absense of extensive results, we present additional evaluation against baselines that do not provide source codes or pre-trained models in the supplementary, using a small set of results obtained from the corresponding authors.

3.2.1 Quantitative Evaluation

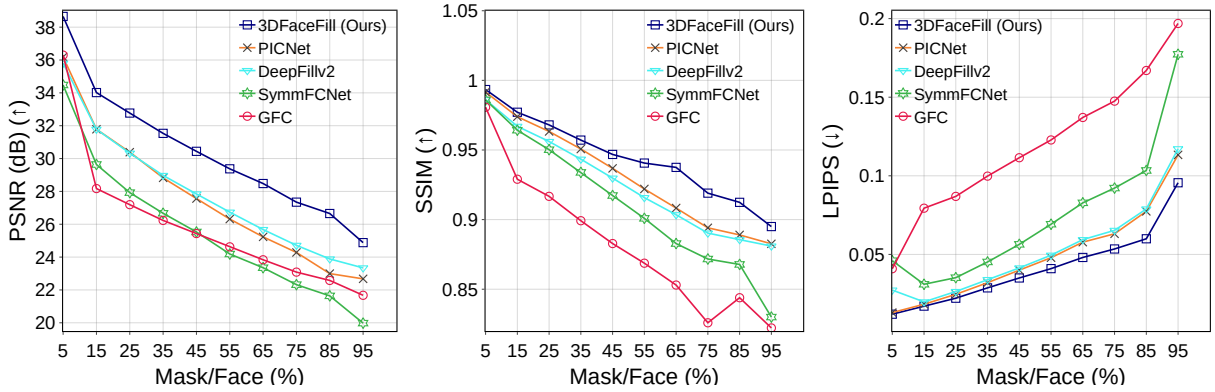
In addition to the typically used PSNR and SSIM metrics, we report LPIPS [Zhang et al., 2018b], which is more suitable for image completion. Table 3.1 reports the overall values of these

[1]Source: https://unsplash.com/s/photos/face

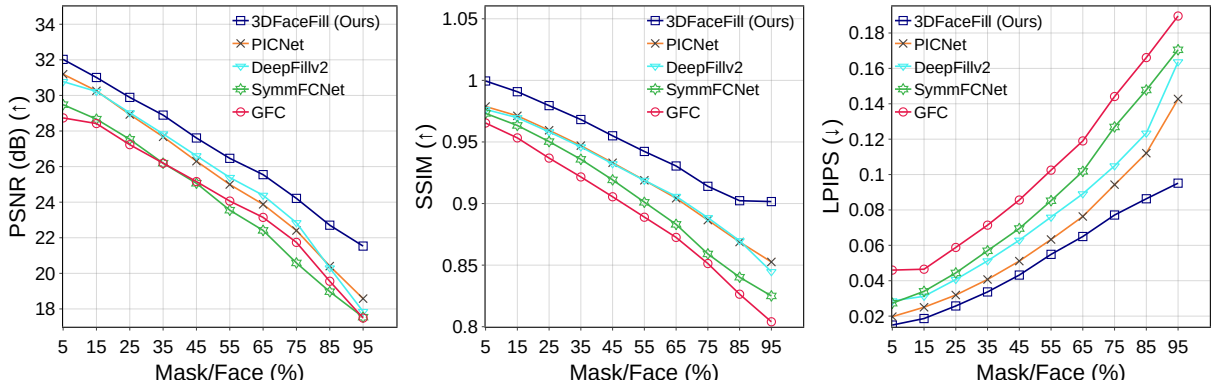
[2]Following author guidelines, we sample top 10 completions ranked by its discriminator and chose the one closest to the groudtruth for evaluation.



(a) CelebA dataset [Liu et al., 2015]



(b) CelebA-HQ dataset [Lee et al., 2020]



(c) MultiPIE dataset [Gross et al., 2010]

Figure 3.3 Quantitative Evaluation: We perform face completion over (a) CelebA, (b) CelebA-HQ and (c) MultiPIE datasets across a range (0-90%) of mask to face area ratios and evaluate the PSNR, SSIM and LPIPS [Zhang et al., 2018b] metrics. Our proposed 3DFaceFill consistently outperforms all the baselines across all the datasets and mask-to-face area ratios.

Dataset	Metric	GFC [Li et al., 2017b]	SymmFC [Li et al., 2020a]	DeepFillv2 [Yu et al., 2019]	PIC [Zheng et al., 2019a]	3DFaceFill
CelebA	**PSNR** (↑)	27.0298	25.8817	28.2097	28.1262	**30.4917**
	SSIM (↑)	0.9257	0.9273	0.9356	0.9424	**0.9521**
	LPIPS (↓)	0.1134	0.0537	0.0499	0.0362	**0.0326**
CelebAHQ	**PSNR** (↑)	25.5836	25.6203	27.9885	27.7020	**29.9398**
	SSIM (↑)	0.8895	0.9232	0.9311	0.9380	**0.9492**
	LPIPS (↓)	0.1076	0.0535	0.0394	0.0376	**0.0365**
MultiPIE (Pose)	**PSNR** (↑)	24.7557	24.7177	26.3385	26.4301	**27.8226**
	SSIM (↑)	0.9187	0.9289	0.9383	0.9451	**0.9482**
	LPIPS (↓)	0.0822	0.0692	0.0527	0.0471	**0.0409**
MultiPIE (Illu)	**PSNR** (↑)	23.5749	24.4813	26.4981	26.2938	**27.8865**
	SSIM (↑)	0.8676	0.8618	00.8718	0.8825	**0.8935**
	LPIPS (↓)	0.1232	0.0747	0.0640	0.0540	**0.0484**
Internet	**PSNR** (↑)	24.1775	24.2829	26.4957	25.6326	**28.8463**
	SSIM (↑)	0.9042	0.9168	0.9293	0.9317	**0.9526**
	LPIPS (↓)	0.0913	0.0625	0.0493	0.0466	**0.0390**

Table 3.1 Quantitative evaluation of face-completion across the CelebA [Liu et al., 2015], CelebAHQ [Lee et al., 2020], subset of MultiPIE [Gross et al., 2010] with pose variations, subset of MultiPIE with illumination variations and internet downloaded in-the-wild images (Internet) datasets (averaged over all mask-to-face ratios). Our method performs significantly better than other approaches in terms of PSNR, SSIM and LPIPS [Zhang et al., 2018b].

metrics across all image-mask pairs for each dataset. Overall 3DFaceFill improves PSNR by 2dB-3dB and LPIPS by 5-10% over the closest baselines. In addition, for all the methods, we report PSNR, SSIM and LPIPS as a function of mask to face area ratio ($\frac{\#MaskPixels}{\#FacePixels}$) in Fig. 3.3a, 3.3b and 3.3c for the CelebA, CelebA-HQ and Multi-PIE datasets, respectively. For the CelebA dataset, we also show the error bands for each method. We make the following observations: (1) Across all the datasets, 3DFaceFill achieves significantly better PSNR and LPIPS across all mask ratios. (2) As can be seen from the error bands in Fig. 3.3a, the worst face completions by 3DFaceFill are better than the best completions from most baselines. (3) Among the baselines, PIC [Zheng et al., 2019a] and DeepFillV2 [Yu et al., 2019] perform comparably with the former being slightly better in terms of LPIPS. (4) The effectiveness of 3DFaceFill over the baselines is more apparent as larger parts of the face are to be completed i.e., as the mask ratio increases. (5) On the CelebA dataset

	PSNR (↑)	**SSIM** (↑)	**LPIPS** [Zhang et al., 2018b] (↓)
DSA [Zhou et al., 2020a]	28.6205	0.9375	0.0436
PConv [Liu et al., 2018]	29.3067	0.9479	0.0379
3DFaceFill	**31.8823**	**0.9615**	**0.0335**

Table 3.2 Quantitative comparison of the proposed 3DFaceFill *vs.* PConv [Liu et al., 2018] and DSA [Zhou et al., 2020a] on a small set of completed images obtained from the authors.

[Liu et al., 2015], the improvement ranges from ∼2dB PSNR for 0-10% mask ratio to ∼4dB PSNR for 60-80% mask ratio. In terms of LPIPS, the improvement ranges from 5% for 0-10% mask ratio to 25% for 60-90% mask ratio. Similar trends are seen across the CelebA-HQ [Lee et al., 2020] and MultiPIE [Gross et al., 2010] datasets too. These results confirm our hypothesis that explicitly modeling the image formation process leads to significantly better face completion. We provide addtional quantitative comparisons against PConv [Liu et al., 2018], DSA [Zhou et al., 2020a] and UVGAN [Deng et al., 2018] in the supplementary since these results are based on a limited number of author-provided completions in the absense of source codes.

3.2.2 Qualitative Evaluation

Figs. 3.4 and 3.5 qualitatively compare face completion between 3DFaceFill and the baselines over a wide variety of challenging conditions. Completions by the baselines are less photorealistic and often contain artifacts in scenarios with dark complexion, tend to deform facial components (*e.g*. nose) and fail to preserve symmetry (*e.g*. eye-gaze or eye-brow shape). In addition, the baselines tend to deform the shape of small faces (*e.g*. children) since they are mostly trained on adult faces where the relative proportions of facial parts differs significantly. In contrast, 3DFaceFill generates more photorealistic completions in all these cases (diverse conditions and mask types) due to explicit 3D shape modeling, incorporating symmetry priors and disentanglement of pose and illumination.

3.2.3 Comparison against PConv and DSA

PConv [Liu et al., 2018] and DSA [Zhou et al., 2020a] have not released publicly available source codes or pre-trained models. Hence, to compare against them, we obtained face completions

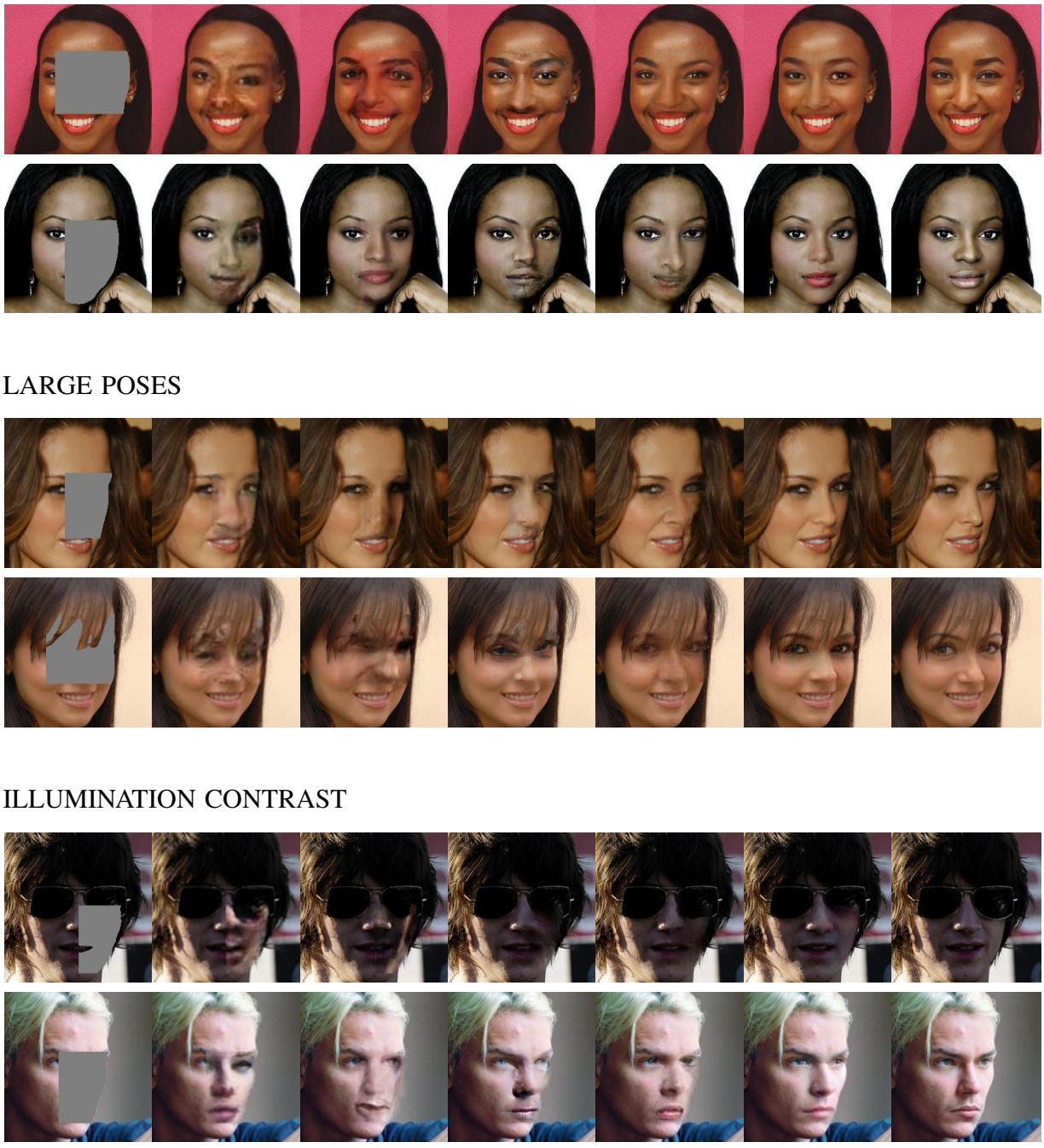
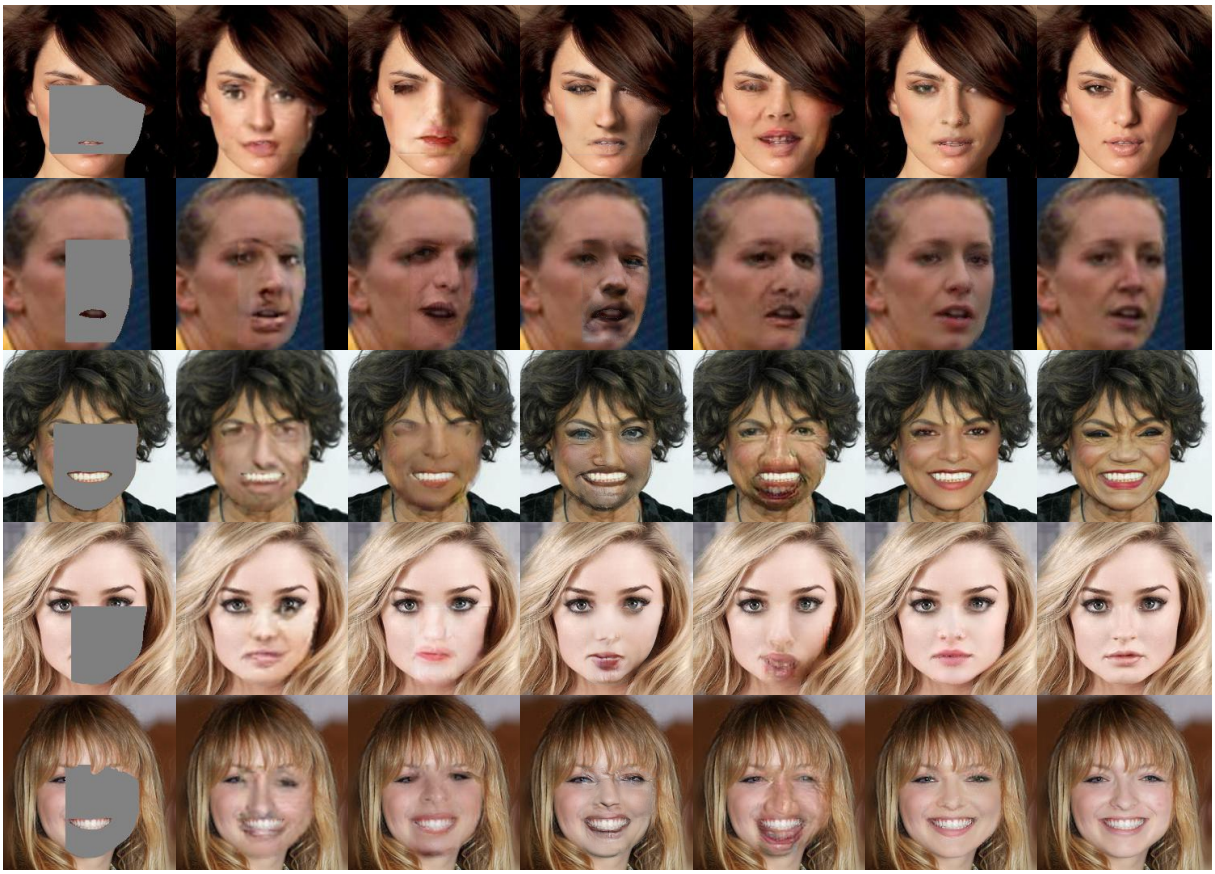


Figure 3.4 Qualitative evaluation under diverse conditions (complexion, pose, illumination).

ASYMMETRY IN EYE-GAZE

SHAPE DEFORMATIONS



Input | GFC [Li et al., 2017b] | SymmFC [Li et al., 2020a] | DeepFillv2 [Yu et al., 2019] | PIC [Zheng et al., 2019a] | 3DFaceFill (Ours) | Ground Truth

Figure 3.5 Qualitative evaluation under diverse conditions (eye-gaze, shape).

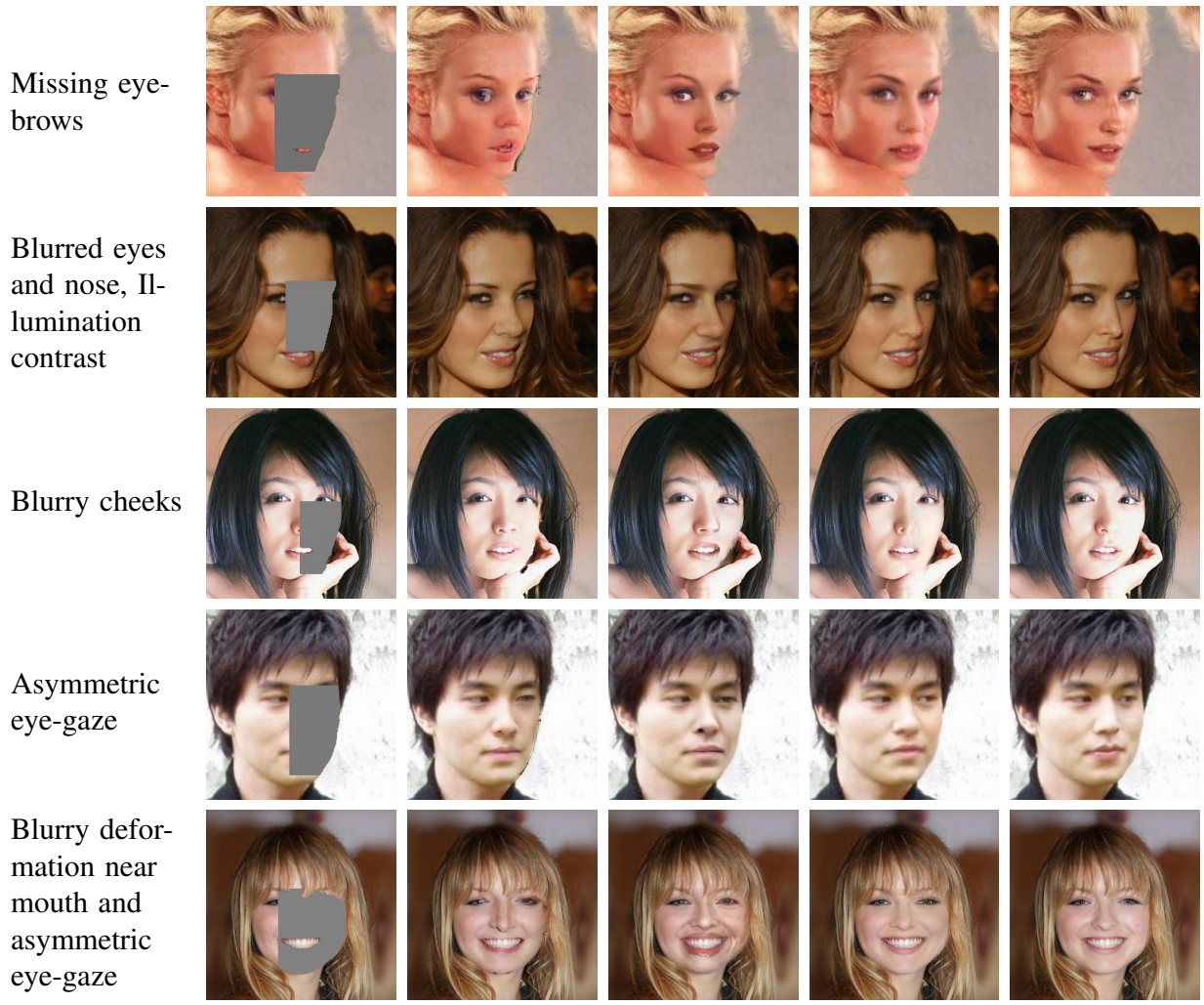


Figure 3.6 Qualitative evaluation of 3DFaceFill *vs.* PConv [Liu et al., 2018] and DSA [Zhou et al., 2020a] on a subset of images received from the respective authors. The text on the left mention the specific deformities in the baselines (blurriness, artifacts, asymmetry and other geometric deformations), that is not present in the completions by 3DFaceFill.

for a small set of 14 partial images through correspondence with the respective authors[3]. We show qualitative results in Fig. 3.6. One can observe that while PConv [Liu et al., 2018] and DSA [Zhou et al., 2020a] tend to deform the facial components under certain conditions leading to geometric and photometric artifacts, 3DFaceFill is free of such artifacts and generates more realistic completions. In addition, we provide quantitative metrics on this small set in Tab. 3.2, where 3DFaceFill reports better PSNR, SSIM and LPIPS [Zhang et al., 2018b] metrics over both the baselines.

3.2.4 Cross-Dataset Evaluation

To further demonstrate the improved generalization performance and robustness afforded by our method, we perform a cross-dataset comparison on the pose and illumination varying images from the MultiPIE [Gross et al., 2010] dataset, *using models that were trained on the CelebA dataset [Liu et al., 2015].* Note that most baselines [Yu et al., 2018, Li et al., 2017b, Zheng et al., 2019a, Zhou et al., 2020a] do not perform such an evaluation. We split the MultiPIE [Gross et al., 2010] dataset into two subsets: (1) a pose varying subset with constant frontal illumination and expression, referred to as *MultiPIE:Pose* and (2) an illumination varying subset with constant frontal pose and expression, referred to as *MultiPIE:Illu*. Table 3.1 reports the PSNR, SSIM and LPIPS [Zhang et al., 2018b] metrics for all the methods on these two splits. It can be seen that 3DFaceFill significantly outperforms the baselines in both the splits. Further, we show qualitative results by 3DFaceFill *vs.* the baselines DeepFillv2 [Yu et al., 2019] and PIC [Zheng et al., 2019a] in Fig. 3.7 (for Pose) and Fig. 3.8 (for Illumination), respectively. From Fig. 3.7, one can observe that the baselines tend to generate fuzzy and deformed faces for extreme poses while 3DFaceFill generates sharper and geometry-preserving completions. And, in the illumination-varying case, DeepFillv2 [Yu et al., 2019] tends to generate artifacts and PIC [Zheng et al., 2019a] tends to generate asymmetric completions for extreme illumination, whereas the completions by 3DFaceFill are free of such artifacts and preserve illumination contrast and symmetry.

[3]The images provided by PConv's authors were obtained from a model trained on 512x512 sized images, *vs.* 256x256 for the other baselines including 3DFaceFill.

Figure 3.7 Qualitative evaluation on the MultiPIE:Pose dataset. Image completion by 3DFaceFill *vs.* baselines DeepFillv2 [Yu et al., 2019] and PIC [Zheng et al., 2019a] on the pose-varying MultiPIE:Pose split [Gross et al., 2010]. While the baselines tend to generate blurred and deformed faces in extreme poses, 3DFaceFill is pose-robust and generates more accurate completions across a range of pose.

Figure 3.8 Qualitative evaluation on the MultiPIE:Illu dataset. Image completion by 3DFaceFill *vs.* the baselines DeepFillv2 [Yu et al., 2019] and PIC [Zheng et al., 2019a] on the illumination varying MultiPIE:Illu split [Gross et al., 2010]. While the baselines tend to generate artifacts in extreme illuminations, 3DFaceFill generates completions that look geometrically accurate and preserve the illumination contrast.

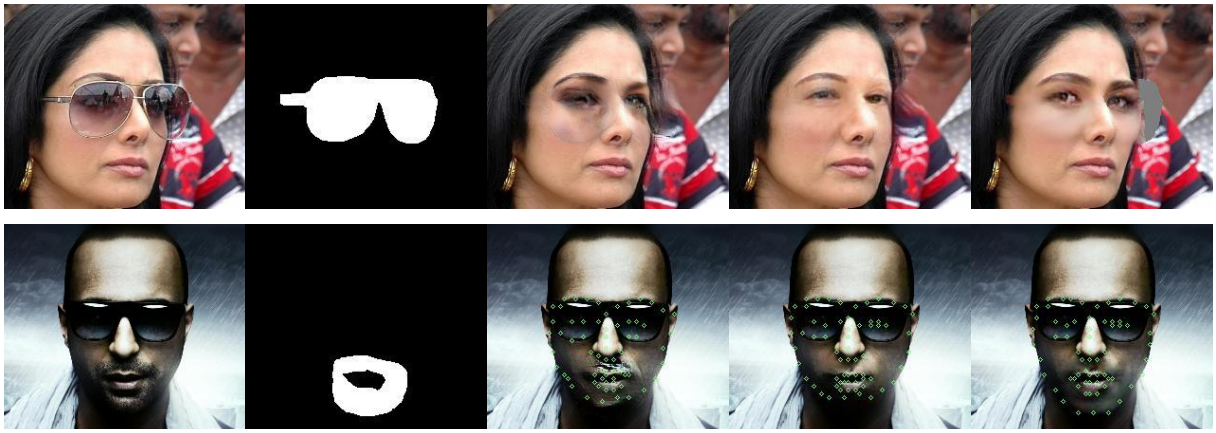


Figure 3.9 Face de-occlusion on real occlusions. The baselines DeepFillv2 and PIC generate non-realistic completion (*e.g.* asymmetric eye-gaze in row 1 and blurry shape in row 2), whereas 3DFaceFill performs realistic de-occlusion, maintaining the structural and photometric integrity of the face.

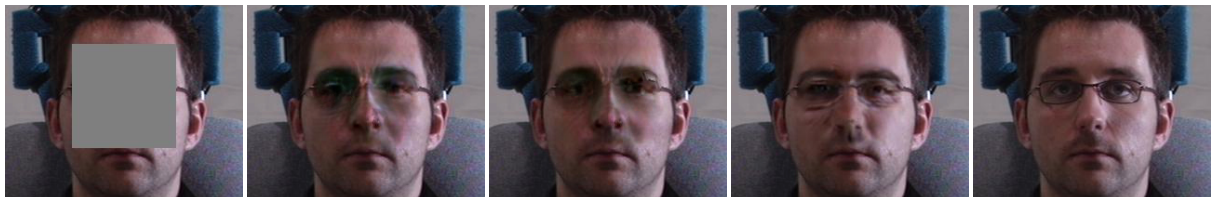
3.2.5 Real Occlusions

One of the potential applications of face completion is in de-occlusion. This is usually challenging when faces have large pose, illumination or shape variations. Fig. 3.9 shows a few real-world de-occlusion examples of faces in such conditions. Notice that, in cases of challenging pose, illumination, *etc*., the baselines tend to generate blurry and asymmetric face completions, whereas 3DFaceFill does more realistic de-occlusion.

3.2.6 Comparison against UVGAN

The proposed face completion method, 3DFaceFill, has three parts, (i) disentangling 2D image into factors such as 3D pose, 3D shape, albedo and illumination (*IL*), (ii) enforcing symmetry in UV albedo (*SYM*), and (iii) iterative refinement of face completion through progressively more accurate 3D pose and shape estimation (*IR*). UVGAN [Deng et al., 2018] on the other hand, (i) performs completion of the missing texture in the UV-representation due to self-occlusion instead

Method	IL	SYM	IR	PSNR (↑)	LPIPS (↓)
UVGAN	✗	✗	✗	28.719	0.0383
UVGAN-*Sym*	✗	✓	✗	28.621	0.0392
3DFaceFill-*NoIR*	✓	✓	✗	29.959	0.0334
3DFaceFill	✓	✓	✓	**30.492**	**0.0326**



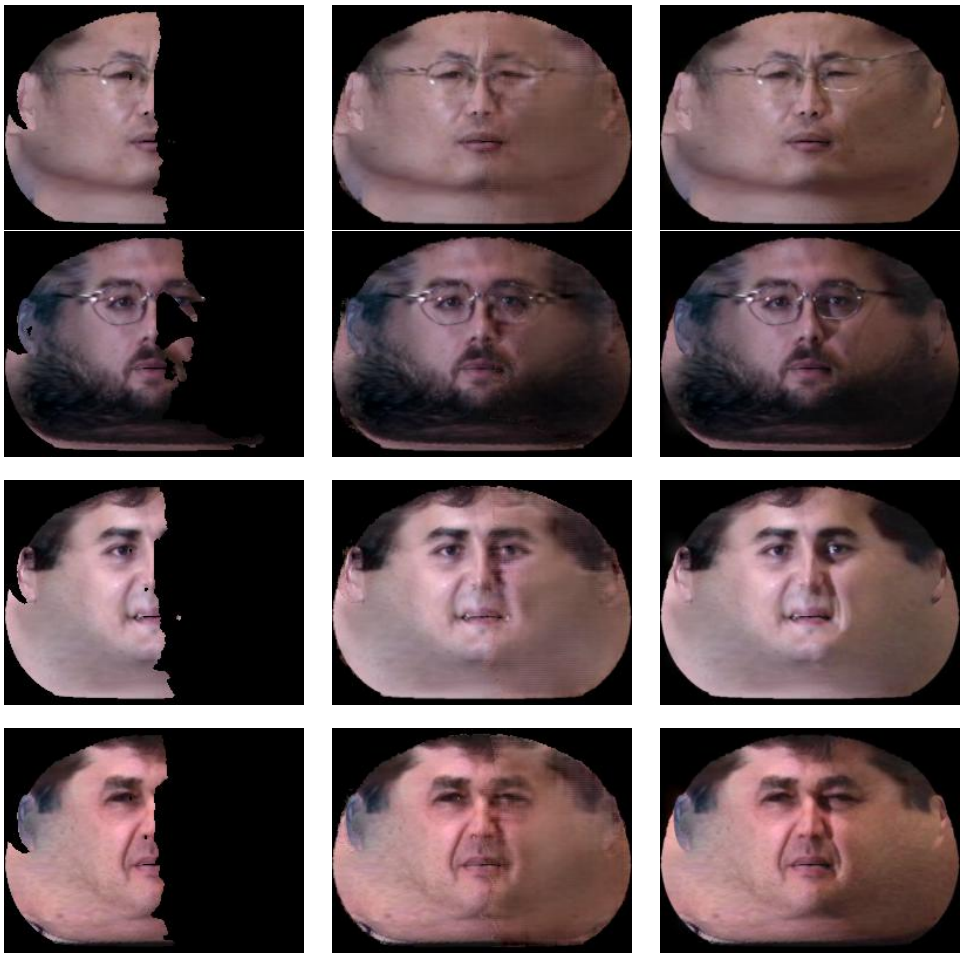
Input UVGAN UVGAN-Sym 3DFaceFill Ground truth

Figure 3.10 Comparing UVGAN [Deng et al., 2018] reformulated for face completion *vs.* 3DFaceFill.

of completing a partial face image itself, (ii) unlike 3DFaceFill, does not disentangle texture further into albedo and illumination, (iii) does not impose symmetry prior on the UV texture, and (iv) uses 3DMM on a fully visible face image rather than a partial image to obtain texture. Since no source code or pretrained model of UVGAN is available, we evaluate these differences in two ways: (A) by reformulating UVGAN for face completion, and (B) comparing UVGAN with our Sym-UNet model on their publicly released texture dataset. We now present the two evaluations.

3.2.7 Comparison with UVGAN [Deng et al., 2018] Reformulated for Face Completion

To simulate UVGAN [Deng et al., 2018] for face completion, we remove the illumination disentanglement (*IL*), symmetry loss (*SYM*) and iterative refinement (*IR*) from 3DFaceFill (refer to Fig. 3.10). We call the variant with *SYM* as UVGAN-*Sym*, and the variant with both *IL* and *SYM* as 3DFaceFill-*NoIR*. Adding *IR* makes for our full model 3DFaceFill. We compare the above-mentioned variants for face completion on the CelebA [Liu et al., 2015] dataset and report the quantitative and qualitative results in Fig. 3.10. One can observe that 3DFaceFill *significantly* outperforms UVGAN as well as the other variants both quantitatively as well as qualitatively. Further, we can see that introducing the symmetry loss (*SYM*) in UVGAN-*Sym* hurts performance since, unlike UV-albedo, UV-texture is not inherently symmetric in faces because of the entangled illumination. Completion on the disentangled albedo (*IL*) instead improves performance in



(a) Input (b) 3DFaceFill (c) Groundtruth

Figure 3.11 Qualitative evaluation of texture completion by the proposed Sym-UNet on the UVDB-MPIE dataset [Deng et al., 2018].

3DFaceFill-*NoIR*. Lastly, iterative refinement (*IR*) further improves completion on top of *IL* and *SYM*. This demonstrates the effectiveness of the novelties that 3DFaceFill introduces over UVGAN [Deng et al., 2018].

3.2.8 Sym-UNet *vs.* UVGAN on Texture Completion

In this evaluation, we trained our Sym-UNet model on the UVDB-MPIE texture dataset released by the authors of UVGAN [Deng et al., 2018]. We split the dataset into a 80:20 train-test split and resized the texture maps to 192×256 for training. Similar to UVGAN, we do not include the symmetry loss because of the presence of illumination variations and the availability of

synthetically completed texture maps, which reduces the utility of symmetry-loss. The rest of the Sym-UNet is retained as such. On the test set, we report a PSNR of 30.1 (*vs.* UVGAN's 25.8) and SSIM of 0.937 (*vs.* UVGAN's 0.886). Further, we show qualitative results in Fig. 3.11, where we see that our completed textures resemble the ground truth closely (we do not have the corresponding completions by UVGAN). Thus, our proposed Sym-UNet network is comparatively better suited for UV-completion than the network used in UVGAN [Deng et al., 2018].

3.2.9 3D View Synthesis of Masked Faces

3DFaceFill has a unique advantage over other face completion approaches, in that unlike existing methods, our method can not only complete partial faces, but also render new views of the completed face from different view-points. In Fig. 3.12, we show this through examples of face views rendered from five different viewpoints by completing the missing albedo and self-occluded regions in the masked faces.

3.2.10 Ablation Studies

3.2.10.1 Effect of Iterative Refinement

To evaluate the effectiveness of iteratively refining face completion at inference, we compare the PSNR, SSIM and LPIPS [Zhang et al., 2018b] metrics on raw output images (before blending with the visible image) at each iteration. As reported in Table 3.3, iteration 2 significantly improves upon iteration 1 over all the metrics. After iteration 2, the metrics become more or less stable, with a slight dip in performance. We hypothesize that it is a result of not training the model for iterative refinement and only performing it at inference. Further, we visualize the absolute difference heatmaps between the completed and the original image for both iterations 1 and 2 in Fig. 3.13 to understand which parts of the face benefit most from refinement. Observe that the largest differences are around the high-detail regions (eyes, beards, *etc*.), which we ascribe to more accurate 3D pose and shape estimation from the completed face after iteration 1 than from the partial face before.

(a) Input (b) Completed and synthesized face views

Figure 3.12 3D Face View Synthesis. 3DFaceFill has the unique ability to not just complete masked faces realistically, but also synthesize new views from them.

	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 6
PSNR (↑)	33.7587	**34.5347**	34.5018	34.4943	34.4428	34.4018
SSIM (↑)	0.9510	**0.9678**	0.9675	0.9670	0.9666	0.9652
LPIPS (↓)	0.0192	**0.0185**	0.0186	0.0187	0.0188	0.0188

Table 3.3 Quantitative evaluation of iterative refinement.

Metric	Full GAN	Patch GAN	NoSym	NoSym+Attn	Full Model
PSNR (↑)	31.7125	31.7552	31.6110	31.7969	**32.1950**
SSIM (↑)	0.9654	0.9658	0.9665	0.9667	**0.9678**
LPIPS (↓)	0.0462	0.0454	0.0446	0.0442	**0.0410**

Table 3.4 Quantitative evaluation between the different ablation models and our full model on masks blocking one-half of the face.

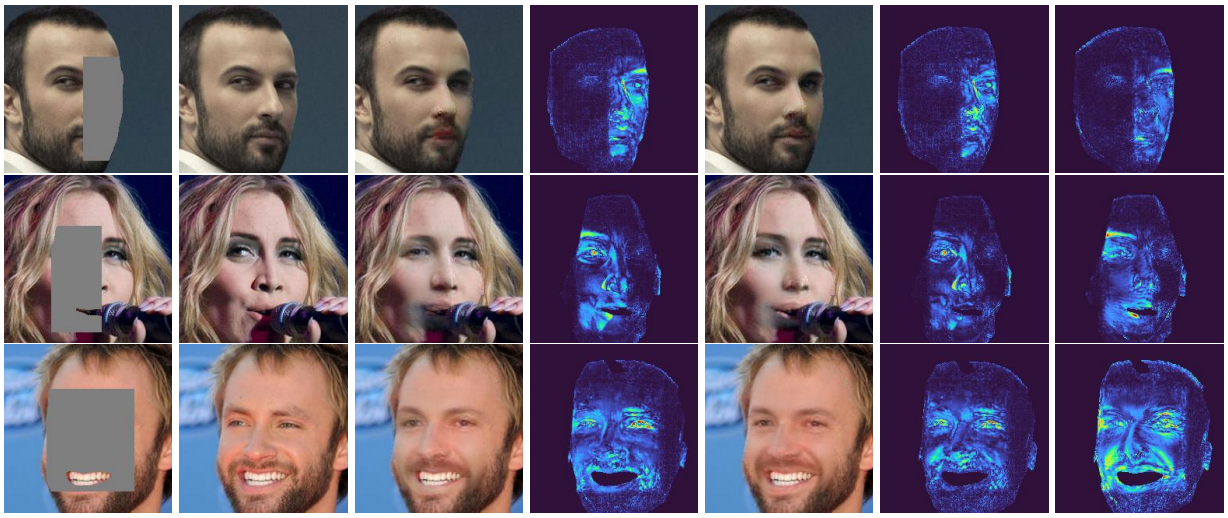


Figure 3.13 Effect of Iterative Finetuning. We show raw completions (without blending) at iterations 1 and 2 along with the difference heatmaps. Note the improvements in *Iter2* over *Iter1* and the corresponding heatmap activations around eyes, eye-brows and other edges on the face.

Figure 3.14 Effect of using Symmetry. The full model includes Sym-UNet and symmetry loss (during training) and can copy symmetric features when available. The absolute difference heatmaps (Full-NoSym) shows that most difference is coming from components such as eyes, eye-brows, *etc*.

3.2.10.2 Effect of Symmetry Constraint

To evaluate the effectiveness of Sym-UNet and the symmetry loss, we compare two variants of the full model (Sym-UNet + symmetry loss). These include, (1) **NoSym:** Sym-UNet replaced by standard UNet and with no symmetry loss, and (2) **NoSym+Attn:** NoSym model plus a self-attention layer after the 3rd upsampling layer in the UNet decoder. Attention layers are commonly employed by many inpainting models [Yu et al., 2018, Yu et al., 2019, Zheng et al., 2019a] for capturing long-range spatial dependencies, so this variant seeks to compare the utility of attention in lieu of symmetry priors for face inpainting. To best evaluate the benefit of symmetry constraints for faces, the above model variations are evaluated on face images masked on one side of the face as shown in Fig. 3.14.

The results in Table 3.4 indicate that the full model outperforms all the variants, with NoSym being the worst among them. Also the NoSym+Attn variant does perform slightly better than NoSym but is still far behind the full model. This indicates that, (i) though attention helps in the absence of any prior constraints, explicitly enforcing geometric priors associated with structured objects like faces is significantly more effective than implicitly learning them through attention, and (ii) symmetry is a more useful feature for face inpainting and behaves like an attention on the visible symmetric parts. As shown in Fig. 3.14, compared to the full model, the NoSym variant results in larger inpainting errors as indicated by the difference heatmaps. Therefore, unlike the full model the NoSym model tends to ignore the visible symmetric regions of the face leading to inconsistencies between the visible and inpainted regions.

3.2.10.3 Effect of Symmetry Gating

We visualize the intermediate gating maps used in our model that control the flow of information in the network (ref Fig. 3.15). We visualize two (out of 64) gating activations (1st - Gate1 and 33rd - Gate2) from the second layer of our Sym-UNet network. As can be seen in Fig. 3.15, while Gate1 activates for the visible regions in the input albedo, Gate2 activates for the masked regions to propagate useful features from the horizontally flipped albedo map to the symmetric side. This enables Sym-UNet to leverage and maintain facial symmetry for inpainting. We also visualize the

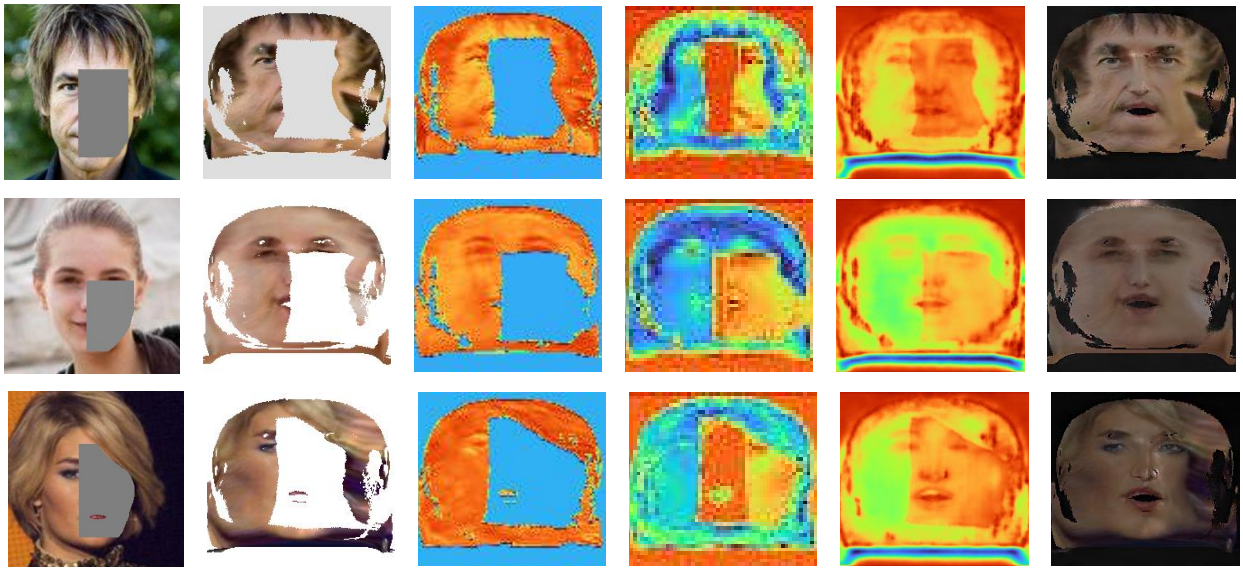


Figure 3.15 Visualizing the Gating Activations and the Uncertainty-Maps. Observe that, while *Gate 1* activates for the visible regions, *Gate 2* activates for the masked regions to propagate useful features from the visible symmetric parts to their masked counterparts. The uncertainty map captures the model's uncertainty around the masked regions and the facial components such as the eyes, thus incurring higher losses for these regions. (*Note:* higher values are represented by warmer (redish) colors in the gating and uncertainty heatmaps).

estimated uncertainty map (σ) in Fig. 3.15 that is learned by the inpainter $\mathcal{G}$ in an unsupervised way. Note that the uncertainty is usually higher around important facial components like the eyes and the masked regions, which increases the loss incurred in these regions.

3.2.11 Discussions

The above described experiments and ablation studies demonstrate the effectiveness of 3DFace-Fill, along with the utility of each of its components in performing robust face completion in challenging cases of facial pose, shape, illumination, *etc*. However, the formulation of our proposed approach do impose a dependency on the fidelity of the underlying 3D model. Essentially, our approach cannot inpaint on regions which are not included in the underlying 3D model and the resolution of inpainting depends on the density of the 3D mesh. 3DFaceFill currently uses the BFM model [Paysan et al., 2009], thanks to its widespread support. However, BFM [Paysan et al., 2009] does not include the inner mouth, hairs and the upper head and has limited vertex density around the eyes, which restricts inpainting in these regions. However, these limitations of the un-

derlying 3D model are not inherent to the proposed approach and do not invalidate the advantages of our model in improving the geometric and photometric consistency of completion. Furthermore, these limitations can potentially be mitigated by substituting BFM with a more detailed 3D face model, such as the Universal Head Model (UHM) [Ploumpis et al., 2020], that includes the inner mouth and detailed eye-balls, along with other improvements.

3.3 Conclusion

In this chapter, we proposed 3DFaceFill, a 3D-aware face completion method. Our solution was driven by the hypothesis that performing face completion on the UV representation, as opposed to 2D pixel representation, will allow us to effectively leverage the power of 3D correspondence and ultimately lead to face completions that are geometrically and photometrically more accurate. Experimental evaluation across multiple datasets and against multiple baselines show that face completions from 3DFaceFill are significantly better, both qualitatively and quantitatively, under large variations in pose, illumination, shape and appearance. These results validate our primary hypothesis.

CHAPTER 4

COLA-SDF: CONTROLLABLE LATENT STYLESDF FOR DISENTANGLED 3D FACE GENERATION

Face generation has a long history in the vision and graphics communities. The earliest of these were based on 3D morphable models (3DMMs) [Paysan et al., 2009, Gerig et al., 2018, Li et al., 2017a]. These models are highly controllable and allow editing of features such as shape, expression, texture, pose, and illumination in a disentangled manner. However, as they are linear models based on principal components analysis (PCA), the faces synthesized by these models lack fine details in shape and appearance. To address this, there has been a growth in nonlinear 3D face reconstruction approaches [Medin et al., 2022, Feng et al., 2021, Tran and Liu, 2018]. These nonlinear approaches have significantly improved the expressivity of 3DMM models but are still far behind the image quality generated by generative adversarial networks (GANs). The strict correspondence assumption is one of the core limitations in terms of modelling for 3DMMs. On the one hand, it simplifies modeling drastically, but it limits the ability to model texture, hair and other elements that lack correspondence.

The striking photorealism of 2D style-based GANs [Karras et al., 2019, Karras et al., 2020, Karras et al., 2021], as well as the ability of implicit neural representations [Mildenhall et al., 2020] to learn detailed 3D object representations from 2D images, have led researchers to combine the benefits of both models. The combined models [Gu et al., 2021, Or-El et al., 2022], often referred to as implicit 3D-GANs, can be trained in an unsupervised way to learn and synthesize the 3D structure and high-fidelity texture of faces. Essentially, implicit 3D-GANs learn to generate an implicit representation of a 3D scene that can be rendered using volumetric rendering similar to that in [Mildenhall et al., 2020]. Unlike both linear and nonlinear 3DMMs, highly complex structures that do not follow the correspondence assumption (such as hair) can be part of the model. However, existing implicit 3D-GANs are not able to support disentangled control or editing of physical attributes such as shape, pose, albedo and illumination, and they require complicated inversion-based approaches to perform such editing with limited success [Xia et al., 2022].

The main idea of our proposed model is imparting controllable generation and editing to implicit 3D-GANs. Previous methods [Tewari et al., 2020b, Medin et al., 2022] have combined the photorealism of 2D GANs with the controllability of 3DMMs with good success, but both methods suffer from limitations. Because StyleRig [Tewari et al., 2020b] relies on the pretrained StyleGAN, its disentangled controllability is limited to the amount of disentanglement in the pretrained StyleGAN; for example, the inherently 2D nature of StyleGAN hampers its disentanglement of pose from other attributes. MOST-GAN, a nonlinear 3DMM in which the texture map is modeled using the StyleGAN2 architecture, is excellent at modeling the 3D shape and texture of faces, but it is unable to model the hair region in full 3D as there is no point-to-point correspondence across subjects in the hair region.

By combining the ability of StyleSDF [Or-El et al., 2022] to learn 3D generation from 2D images with the disentangled controllability of MOST-GAN [Medin et al., 2022], we can retain the best features of both implicit 3D-GANs and nonlinear 3DMMs. By incorporating the nonlinear 3DMM via loss functions only, we maintain the photorealism provided by the StyleSDF architecture. The control is enforced during training of the StyleSDF architecture via loss functions that incorporate MOST-GAN's disentangled parameters using inverse rendering with MOST-GAN's image decoder.

To summarize, in this chapter we propose CoLa-SDF, which imparts controlled face generation and editing to implicit 3D-GAN. Our proposed approach utilizes a differentiable nonlinear 3DMM-based model to supervise the training of an implicit 3D-GAN in order to learn disentangled representations for shape, texture, and illumination. In addition, we employ face parsing (semantic segmentation of face images) to further disentangle a latent code for the hair and background from the latent representation of the face. As a result, CoLa-SDF can generate high-fidelity 3D faces, which can then be edited by changing separate latent codes for shape, texture, illumination, pose, and hair and background, either independently or in various combinations. In summary, our main contributions include:

- We propose a new method called CoLa-SDF that allow generation and subsequent editing

of high-fidelity 3D faces, from which photorealistic 2D images can be rendered in multiple views.

- Our method builds upon the architecture of StyleSDF while disentangling the latent representation into separate latent codes for shape, texture, illumination, and hair and background, thereby allowing independent editing of each attribute.

- To achieve disentangled control, we sample in the PCA space of MOST-GAN parameters and introduce novel parametric and image-based consistency losses utilizing the MOST-GAN encodings and face parsing.

4.1 Preliminaries

Our method relies on StyleSDF [Or-El et al., 2022] and MOST-GAN [Medin et al., 2022], which we now introduce in more detail.

StyleSDF [Or-El et al., 2022] consists of two components: a signed distance function (SDF)-based volume renderer and a styled generator. Given a latent code $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$, the volume renderer takes in a 3D query point $\mathbf{x}$ and a viewing direction $\mathbf{v}$ and maps them into an SDF value $d(\mathbf{x}, \mathbf{z})$, a radiance $c(\mathbf{x}, \mathbf{v}, \mathbf{z})$, and a feature vector $f(\mathbf{x}, \mathbf{v}, \mathbf{z})$. A low-resolution (64×64) image $\mathbf{I}_{\text{vol}}$ and feature map $\mathbf{F}$ are generated using volume rendering. Each pixel is computed by querying points along the ray $\mathbf{r} = \mathbf{o} + t\mathbf{p}$ originating from the camera position $\mathbf{o}$ and passing through the pixel location corresponding to $\mathbf{p}$ as follows:

$$\begin{aligned} \mathbf{I}_{\text{vol}} &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{p})dt, \\ \mathbf{F}(\mathbf{r}) &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{f}(\mathbf{r}(t), \mathbf{p})dt, \end{aligned} \tag{4.1}$$

where $T(t) = \exp\left(-\int_{t_n}^{t_f} \sigma(\mathbf{r}(s))ds\right)$ represents the visibility of each point along the ray. The density field $\sigma(\mathbf{x})$ is obtained from the SDF $d(\mathbf{x})$ using the following model:

$$\sigma(\mathbf{x}) = \frac{1}{\delta}\text{Sigmoid}\left(\frac{-d(\mathbf{x})}{\delta}\right), \tag{4.2}$$

where δ is a learned parameter. The styled generator maps the feature map $\mathbf{F}$ into a high-resolution image $\mathbf{I}$ conditioned on the style-code $\mathbf{w} = g(\mathbf{z})$.

The volume renderer and the styled generator are trained separately. First, the volume renderer is trained along with a low-resolution discriminator in an adversarial way. Then, the volume renderer's weights are frozen, and the styled generator is trained in an adversarial way, along with a high-resolution discriminator. The volume renderer loss $\mathcal{L}_{\text{vol}}$ consists of the non-saturating GAN loss with R1 regularization [Mescheder et al., 2018] $\mathcal{L}_{\text{adv}}$, pose alignment loss $\mathcal{L}_{\text{view}}$, eikonal loss $\mathcal{L}_{\text{eik}}$, and minimal surface loss $\mathcal{L}_{\text{surf}}$:

$$\mathcal{L}_{\text{vol}} = \mathcal{L}_{\text{adv}} + \lambda_{\text{view}}\mathcal{L}_{\text{view}} + \lambda_{\text{eik}}\mathcal{L}_{\text{eik}} + \lambda_{\text{surf}}\mathcal{L}_{\text{surf}}, \tag{4.3}$$

where $\lambda_{\text{view}}, \lambda_{\text{eik}}$, and λ_{surf} are the weights for the pose, eikonal, and minimal surface losses, respectively.

The pose alignment loss $\mathcal{L}_{\text{view}}$ enforces that the generated images follow the input pose. This loss is applied both on the volume generator, as well as the low-resolution discriminator (but only when iterating through generated images). For this, the low-resolution disciminator is modified, such that, in addition to the image score, it also predicts the pose $(\hat{\phi}, \hat{\theta})$ of the image. The pose alignment loss is defined as the smoothed L1 loss between the pose (ϕ, θ) used by the volume renderer to generate images, and the pose $(\hat{\phi}, \hat{\theta})$ predicted by the low-resolution discriminator:

$$\mathcal{L}_{\text{view}} = \begin{cases} (\hat{\theta} - \theta)^2 & \text{if } |\hat{\theta} - \theta| \leq 1 \\ |\hat{\theta} - \theta| & otherwise \end{cases} \tag{4.4}$$

The eikonal loss enforces physical validity of the signed distance field [Gropp et al., 2020]:

$$\mathcal{L}_{\text{eik}} = \mathbb{E}_{(\mathbf{x}} \left(||\nabla d(\mathbf{x})||_2 - 1 \right)^2 . \tag{4.5}$$

The minimal surface loss penalizes the SDF values that are close to zero to avoid spurious zero-crossings and non-visible surfaces from being formed:

$$\mathcal{L}_{\text{surf}} = \mathbb{E}_{(\mathbf{x}} \left(exp(-100|d(\mathbf{x})|) \right) . \tag{4.6}$$

The styled generator is trained using a combination of a path regularization loss $\mathcal{L}_{\text{path}}$ as well as $\mathcal{L}_{\text{adv}}$ defined above:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{adv}} + \lambda_{\text{path}}\mathcal{L}_{\text{path}}, \tag{4.7}$$

where λ_{path} is the weight of the path loss.

MOST-GAN [Medin et al., 2022] is a nonlinear 3DMM that includes a set of encoders for shape $\mathcal{E}_\alpha$, albedo $\mathcal{E}_\tau$, illumination $\mathcal{E}_\gamma$, and pose $\mathcal{E}_\theta$, a shape decoder $\mathcal{G}_\alpha$ and an albedo decoder $\mathcal{G}_\tau$. Given a face image, the encoders extract the shape parameters $\boldsymbol{\alpha}$, the albedo parameters $\boldsymbol{\tau}$, the spherical harmonics illumination parameters $\boldsymbol{\gamma}$ [Ramamoorthi and Hanrahan, 2001, Zhang and Samaras, 2006] and a 3D pose $\boldsymbol{\theta}$. The decoders generate the full 3D shape $\mathbf{S}$ and albedo map $\mathbf{A}$: $\mathcal{G}_\alpha : \boldsymbol{\alpha} \rightarrow \mathbf{S}$, $\mathcal{G}_\tau : \boldsymbol{\tau} \rightarrow \mathbf{A}$. Next, a differentiable renderer $\mathcal{R}$ [Ravi et al., 2020] renders the reconstructed face image $\mathbf{I}_{\text{most}}$ from the generated 3D model, lighting and pose parameters: $\mathbf{I}_{\text{most}} = \mathcal{R}(\mathbf{S}, \mathbf{A}, \boldsymbol{\gamma}, \boldsymbol{\theta})$. In this work, we use the pre-trained MOST-GAN weights provided by the authors.

4.2 Approach

4.2.1 Overview

Our proposed approach is based on building a semantically disentangled latent space for an implicit 3D GAN, such that each part of the latent code corresponds to a different physical attribute. We achieve this by enforcing a correspondence between the latent codes for these factors (shape, albedo and illumination) and the parameters of a 3DMM, which has built-in disentangled representations of these parameters. Pose control can be easily handled using 3D volume rendering and the view-dependence property of implicit 3D GANs [Gu et al., 2021, Or-El et al., 2022]. However, 3DMMs do not facilitate disentanglement of hair and background, because these attributes are not represented well in 3DMM models. In order to encourage part of the latent code to correspond to hair and background, we introduce a photo-consistency loss on the hair and background regions of the generated images that encourages different faces generated using the same hair and background codes to have consistent hair and background appearance.

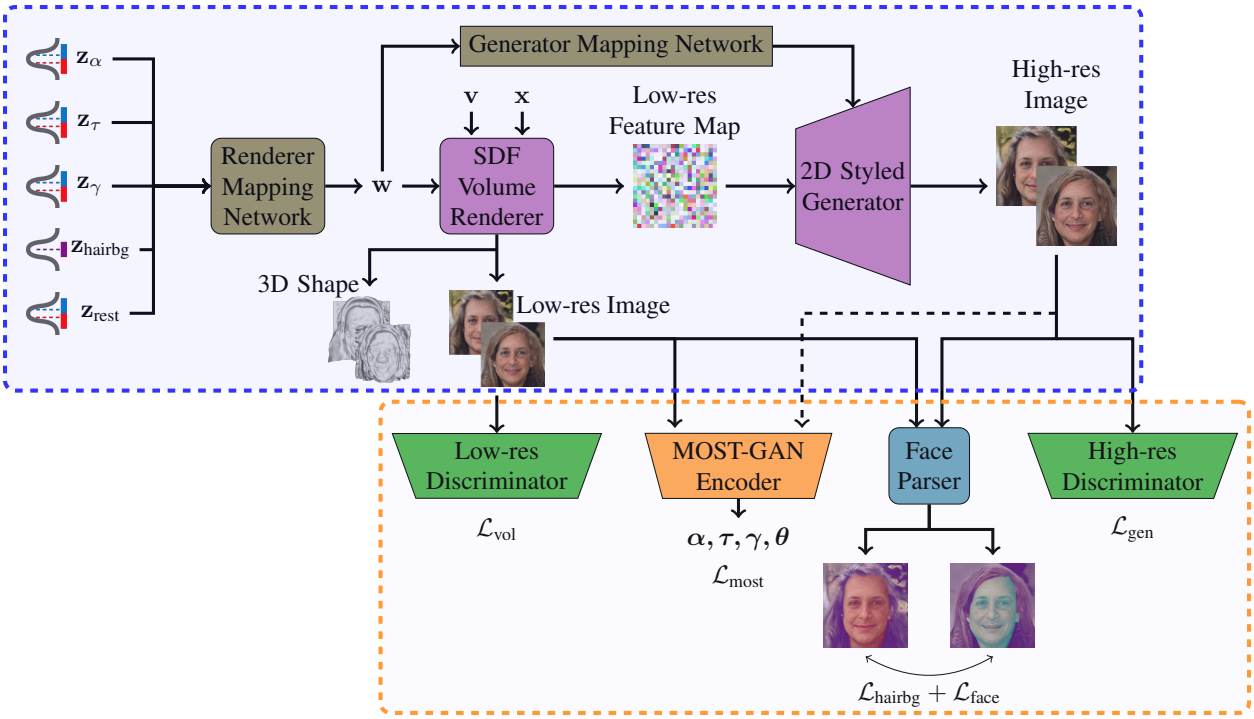


Figure 4.1 Overview: (Top) The SDF volume renderer generates the low-resolution SDF surface, image and feature map conditioned on the latent codes $\mathbf{z}_\alpha$, $\mathbf{z}_\tau$, $\mathbf{z}_\gamma$, $\mathbf{z}_{\text{hairbg}}$ and $\mathbf{z}_{\text{rest}}$, which the styled generator decodes into a high-resolution image. (Bottom) To disentangle shape, albedo and illumination, we enforce parametric consistency between the sampled latent codes and the MOST-GAN encodings $\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\gamma}, \boldsymbol{\theta}$. To disentangle hair/background, we alternately resample face parameters $\mathbf{z}_\alpha, \mathbf{z}_\tau$, and $\mathbf{z}_\gamma$ and enforce image-based consistency on the hair and background; followed by re-sampling $\mathbf{z}_{\text{hairbg}}$ and enforcing consistency on the face regions.

Disentangling the latent space of an implicit 3D GAN according to a 3DMM requires the 3DMM to be differentiable and highly expressive, so for our model we adopted the nonlinear 3DMM model MOST-GAN [Medin et al., 2022], as it matches these requirements. For our implicit 3D GAN architecture, we selected StyleSDF [Or-El et al., 2022], both because of its high rendering quality and because it explicitly models the object's 3D shape in the form of signed distance field (SDF). Since our proposed modifications and enhancements to StyleSDF enable disentangled control of physical attributes by modifying disjoint segments of its latent code, we call our model Controllable Latent StyleSDF (CoLa-SDF).

4.2.2 Architecture

At the core of our method, we use StyleSDF [Or-El et al., 2022] and largely maintain its architecture. In order to successfully disentangle the latent code, we make two key changes to

StyleSDF (refer Fig. 4.1). First, we partition the 256-dimensional latent code $\mathbf{z}$ into separate latent codes that will correspond to the face shape $\mathbf{z}_\alpha$, albedo $\mathbf{z}_\tau$, illumination $\mathbf{z}_\gamma$, and hair and background $\mathbf{z}_{\text{hairbg}}$. We also introduce a final segment of the latent code, $\mathbf{z}_{rest}$, which the model is free to assign to any facial appearance factors not explained by MOST-GAN [Medin et al., 2022]. Second, we modify the training method for StyleSDF and incorporate novel consistency loss functions. One set of consistency loss functions enforces consistency between the latent codes that generate a face and the parameters that MOST-GAN extracts from the generated face image. A second set of consistency loss functions minimizes the impact that changes in $\mathbf{z}_{\text{hairbg}}$ can have on the face appearance, and it similarly minimizes the effect that the face-specific latent codes can have on the hair and background appearance. Careful design of both the latent code factorization and the consistency losses during training are crucial to attain the desired disentanglement. We now describe these in detail.

4.2.3 Latent Code Factorization

We partition the 256 dimensions of the latent code $\mathbf{z}$ into disjoint subsets: 128 dimensions corresponding to the MOST-GAN [Medin et al., 2022] attributes, further partitioned into $\mathbf{z}_\alpha$, $\mathbf{z}_\tau$, and $\mathbf{z}_\gamma$; 64 dimensions $\mathbf{z}_{\text{hairbg}}$ corresponding to hair and background appearance, and 64 dimensions $\mathbf{z}_{rest}$ to account for any remaining details in and around the face. To determine the dimensionality to allot to each of the MOST-GAN factors $\mathbf{z}_\alpha$, $\mathbf{z}_\tau$, and $\mathbf{z}_\gamma$, we perform eigen-decomposition over the corresponding data covariance matrices $\mathbf{\Sigma}_\alpha$, $\mathbf{\Sigma}_\tau$, and $\mathbf{\Sigma}_\gamma$ respectively, that we obtain by encoding images in the FFHQ [Karras et al., 2019] dataset to the MOST-GAN [Medin et al., 2022] shape $\boldsymbol{\alpha}$, albedo $\boldsymbol{\tau}$, and illumination $\boldsymbol{\gamma}$ parameters using the pre-trained encoders. Based on this analysis, we chose a dimensionality of $d_\alpha = 37$ for $\mathbf{z}_\alpha$ and $d_\tau = 64$ for $\mathbf{z}_\tau$, which accounted for well over 95% of the variance in their respective distributions. In order to enable full explicit control over the 27 spherical harmonics lighting parameters used in MOST-GAN, we chose $d_\gamma = 27$ for $\mathbf{z}_\gamma$. Since we desire $\mathbf{z}_\omega \sim \mathcal{N}(\mathbf{0}, I)$ for $\boldsymbol{\omega} \in (\boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\gamma})$, we use eigen-decomposition to create a mapping between the parameter encoding of MOST-GAN [Medin et al., 2022] and the corresponding latent codes in

our model:

$$\boldsymbol{\omega}_{sample} = \boldsymbol{U}'_{\omega}\boldsymbol{\Lambda}'_{\omega}\mathbf{z}_{\omega} + \boldsymbol{\mu}_{\omega}, \tag{4.8}$$

where $\boldsymbol{U}'_{\omega}$ and $\boldsymbol{\Lambda}'_{\omega}$ are the top d_{ω} eigenvectors and eigenvalues of $\boldsymbol{\Sigma}_{\omega}$ and $\boldsymbol{\mu}_{\omega}$ is the data mean.

4.2.4 Training

4.2.4.1 StyleSDF Losses

As in [Or-El et al., 2022], we train the model in two stages. In the first stage, we train the volume renderer, then we freeze its weights in the second stage and train the 2D styled generator. In addition to the original StyleSDF losses, which we described in Sec. 4.1, in both stages we introduce new consistency losses that we will describe in Sec. 4.2.4.2.

In the first stage, training the volume renderer, the loss $\mathcal{L}_{\text{vol}}$ consists of the non-saturating GAN loss with R1 regularization [Mescheder et al., 2018] $\mathcal{L}_{\text{adv}}$, pose alignment loss $\mathcal{L}_{\text{view}}$, eikonal loss $\mathcal{L}_{\text{eik}}$, and minimal surface loss $\mathcal{L}_{\text{surf}}$, as defined in [Or-El et al., 2022]. In the second stage, training the styled 2D generator, the loss $\mathcal{L}_{\text{gen}}$ consists of a path regularization loss $\mathcal{L}_{\text{path}}$ as well as $\mathcal{L}_{\text{adv}}$ defined above:

$$\begin{aligned}
\mathcal{L}_{\text{vol}} &= \mathcal{L}_{\text{adv}} + \lambda_{\text{view}}\mathcal{L}_{\text{view}} + \lambda_{\text{eik}}\mathcal{L}_{\text{eik}} + \lambda_{\text{surf}}\mathcal{L}_{\text{surf}}, \\
\mathcal{L}_{\text{gen}} &= \mathcal{L}_{\text{adv}} + \lambda_{\text{path}}\mathcal{L}_{\text{path}},
\end{aligned} \tag{4.9}$$

where $\lambda_{\text{view}} = 15$, $\lambda_{\text{eik}} = 1$, $\lambda_{\text{surf}} = 1$ and $\lambda_{\text{path}} = 2$.

4.2.4.2 CoLa-SDF Losses: MOST-GAN Consistency and Hair/Background Consistency

We introduce the MOST-GAN consistency and hair consistency losses to both stages of training, in addition to the original StyleSDF losses (4.9). In the first stage, our new losses are applied to the low-res images, while in the second stage, they are applied to the high-res images.

We enforce consistency of the rendered image with respect to the sampled MOST-GAN [Medin et al., 2022] parameters using the MOST-GAN consistency loss $\mathcal{L}_{\text{most}}$:

$$\mathcal{L}_{\text{most}} = \lambda_{\alpha}\mathcal{L}_{\alpha} + \lambda_{\tau}\mathcal{L}_{\tau} + \lambda_{\gamma}\mathcal{L}_{\gamma} + \lambda_{\theta}\mathcal{L}_{\theta}, \tag{4.10}$$

where $\mathcal{L}_\alpha = ||\mathbf{E}_\alpha(\mathbf{I}) - \boldsymbol{\alpha}_{sample}||_2^2$ enforces that the MOST-GAN's shape encoding of rendered image $\mathbf{E}_\alpha(\mathbf{I})$ is the same as the sampled shape parameters $\boldsymbol{\alpha}_{\text{sample}}$ obtained from Eq. 4.8. Similarly, we define the albedo consistency loss $\mathcal{L}_\tau$ and the illumination consistency loss $\mathcal{L}_\gamma$ as ℓ_2-error losses between the predicted MOST-GAN parameters and the sampled parameters. We enforce pose-consistency between the pose encodings over the two sub-iterations as $\mathcal{L}_\theta = ||\mathbf{E}_\theta(\mathbf{I}_{s1}) - \mathbf{E}_\theta(\mathbf{I}_{s2})||_2^2$. We set $\lambda_\alpha = 3000, \lambda_\tau = 100, \lambda_\gamma = 100$ and $\lambda_\theta = 1000$.

Existing 3DMM-based approaches do not model hair and background. Hence, to disentangle hair/background from other physical attributes, we adopt a novel approach where we force the hair/background code $\mathbf{z}_{\text{hairbg}}$ to only model the hair and background. Specifically, we perform a second sub-iteration followed by each generator iteration, where, during even iterations, we re-sample $\mathbf{z}_\alpha$, $\mathbf{z}_\tau$ and $\mathbf{z}_\gamma$, and enforce hair and background consistency using $\mathcal{L}_{\text{hairbg}}$. In the odd iterations, we re-sample $\mathbf{z}_{\text{hairbg}}$ and enforce face consistency using $\mathcal{L}_{\text{face}}$. The hair/background and face consistency losses are defined as:

$$\mathcal{L}_{\text{hairbg}} = \mathcal{L}_{\text{photo}}(\mathbf{I}_{s1}, \mathbf{I}_{s2}, \mathbf{M}_h) + \mathcal{L}_{\text{vgg}}(\mathbf{I}_{s1}, \mathbf{I}_{s2}, \mathbf{M}_h) \tag{4.11}$$

$$\mathcal{L}_{\text{face}} = \mathcal{L}_{\text{photo}}(\mathbf{I}_{s1}, \mathbf{I}_{s2}, \mathbf{M}_f) + \mathcal{L}_{\text{vgg}}(\mathbf{I}_{s1}, \mathbf{I}_{s2}, \mathbf{M}_f) \tag{4.12}$$

Here, $\mathbf{I}_{s1}$ and $\mathbf{I}_{s2}$ are the images rendered in sub-iterations 1 and 2, respectively, $\mathbf{M}_h = \mathbf{M}_{\text{hairbg},s1} \cup \mathbf{M}_{\text{hairbg},s2}$ is the union of the *hair* masks from the two sub-iterations, and $\mathbf{M}_f = \mathbf{M}_{\text{face},s1} \cup \mathbf{M}_{\text{face},s2}$ is the union of the *face* masks from the two sub-iterations. We use a pre-trained face parser [Chen et al., 2017a] to parse the rendered face images into one segmentation masks for the face and one for hair and background. We define the masked photometric loss as $\mathcal{L}_{\text{photo}}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{m}) = ||(\mathbf{x}_1 - \mathbf{x}_2) \odot \mathbf{m}||_1$, where $\odot$ is the element-wise product operator. Similarly, we define the masked perceptual loss as $\mathcal{L}_{\text{vgg}}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{m}) = ||\phi(\mathbf{x}_1 \odot \mathbf{m}) - \phi(\mathbf{x}_2 \odot \mathbf{m})||_2^2$.

Thus, the overall loss for stage 1, volume renderer training, is given by:

$$\mathcal{L}_{\text{vol}}^{\text{cola}} = \mathcal{L}_{\text{vol}} + \mathcal{L}_{\text{most}} + \lambda_{\text{hairbg}}\mathcal{L}_{\text{hairbg}} + \lambda_{\text{face}}\mathcal{L}_{\text{face}}. \tag{4.13}$$

Similarly, the overall loss for stage 2, the training of the 2D styled generator, is given by:

$$\mathcal{L}_{\text{gen}}^{\text{cola}} = \mathcal{L}_{\text{gen}} + \mathcal{L}_{\text{most}} + \lambda_{\text{hairbg}}\mathcal{L}_{\text{hairbg}} + \lambda_{\text{face}}\mathcal{L}_{\text{face}}. \tag{4.14}$$

We set $\lambda_{\text{hairbg}} = 5$ in even iterations but $= 0$ in odd iterations, and $\lambda_{\text{face}} = 5$ in odd iterations but $= 0$ in even iterations, for both Eqs. (4.13) and (4.14).

4.2.4.3 Initialization of Each Stage

To obtain meaningful MOST-GAN encodings and face parsing, we need the generated images to look like faces. Hence, we initialize each stage by training with only StyleSDF based losses for up to 5000 iterations, following which $\mathcal{L}_{\text{most}}$, $\mathcal{L}_{\text{hairbg}}$ and $\mathcal{L}_{\text{face}}$ are introduced. Failing to do so may result in longer training time and poor convergence.

4.3 Experiments

4.3.1 Implementation Details

We trained CoLa-SDF's volume renderer and styled generator separately for 400,000 and 200,000 iterations, respectively. We trained the volume renderer with a batch-size of 20 and ray-sampling frequency (samples per ray) of 24, on a machine with Intel Xeon Gold 6326 processor with 64 cores and 10 Nvidia A40 GPUs. While training the styled-generator, we freeze the volume renderer and the renderer mapping network and increase the ray-sampling frequency to 64. We trained the styled-generator with a batch-size of 40 on the same machine. Training the volume renderer takes 3 days and the styled-generator takes 4 days on this machine.

4.3.2 Datasets and Evaluation

We train our model on the FFHQ dataset [Karras et al., 2019], which consists of 70,000 high-resolution images of portrait faces from varying age, ethnicity, and image conditions. We evaluate our model in terms of both its face generation and subsequent editing capabilities. To evaluate generation quality numerically, we compare our model's capability to generate photorealistic images with existing methods in terms of FID. To evaluate image editing, we demonstrate our model's capability to disentangle the latent space for shape, albedo, illumination and hair/background and explicitly edit these properties.

Method	FID (↓)
GRAF [Schwarz et al., 2020]	79.2
PiGAN [Chan et al., 2021]	83.0
GIRAFFE [Niemeyer and Geiger, 2021]	31.2
Ours	**19.4**
StyleSDF [Or-El et al., 2022]	**11.5**

Table 4.1 FID evaluations at 256x256 resolution. Our method, while enabling disentanglement, demonstrates the second best performance.

4.3.3 Face Generation

We demonstrate the face generation capability of CoLa-SDF by rendering face images from multiple viewpoints. Our method's view-consistent synthesis is demonstrated in Fig. 4.2, which renders two randomly generated faces in viewpoints up to ± 0.45 radians azimuth and ± 0.225 radians elevation. To demonstrate the quality of the underlying 3D surface, we also show the corresponding marching cubes mesh obtained from the signed distance field. In addition, for each example, we map the latent code for shape $\mathbf{z}_\alpha$ to the MOST-GAN parameter $\boldsymbol{\alpha}$ using Eq. (4.8) and generate the corresponding MOST-GAN mesh using its decoder $\mathbf{S} = \mathcal{G}_\alpha(\boldsymbol{\alpha})$. As shown in the figure, the generated MOST-GAN meshes correspond well with the images and the marching-cubes mesh generated by our method, which demonstrates that CoLa-SDF has learned a high-degree of correspondence with MOST-GAN.

To quantitatively evaluate image generation quality of our method, we compute the FID [Heusel et al., 2017] metric after downsampling the generated images to a resolution of 256×256. We compare our method against the FIDs reported by GRAF [Schwarz et al., 2020], PiGAN [Chan et al., 2021], GIRAFFE [Niemeyer and Geiger, 2021] and StyleSDF [Or-El et al., 2022]. As shown in Tab. 4.1, while StyleSDF reports the best FID, our method is a close second, a small price to pay for our method's disentangled control over the latent space.

4.3.4 Disentanglement of the Latent Space

While most 3DMM-based models can only disentangle shape, albedo, and illumination, our model additionally provides separate control over hairstyle and background. In the following sub-sections, we qualitatively and quantitatively evaluate CoLa-SDF's latent space disentanglement in

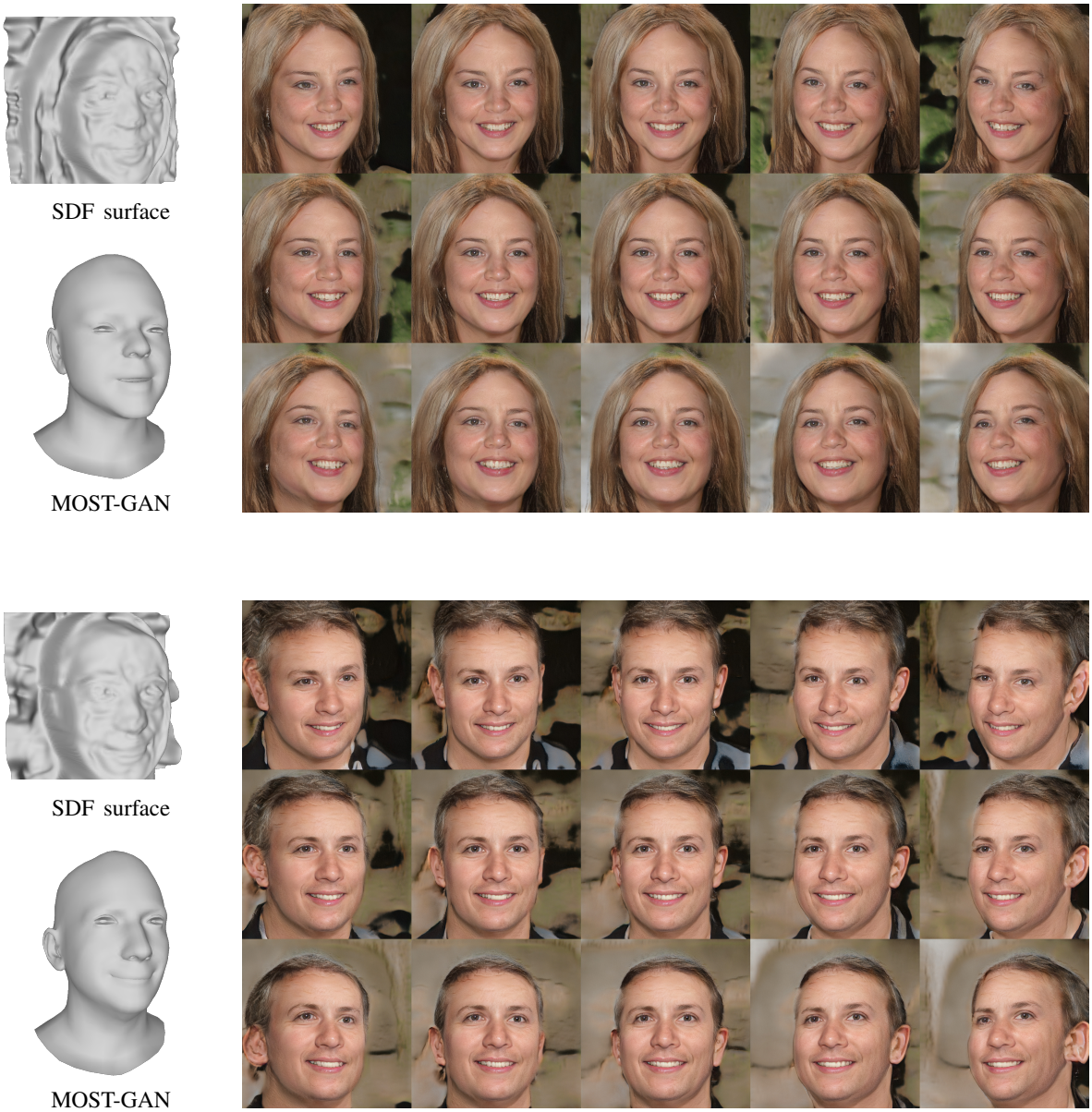


Figure 4.2 Multiview image renderings and 3D shapes extracted from SDF from CoLa-SDF, along with the corresponding MOST-GAN [Medin et al., 2022] reconstructions.

terms of shape, albedo, illumination, and hair/background.

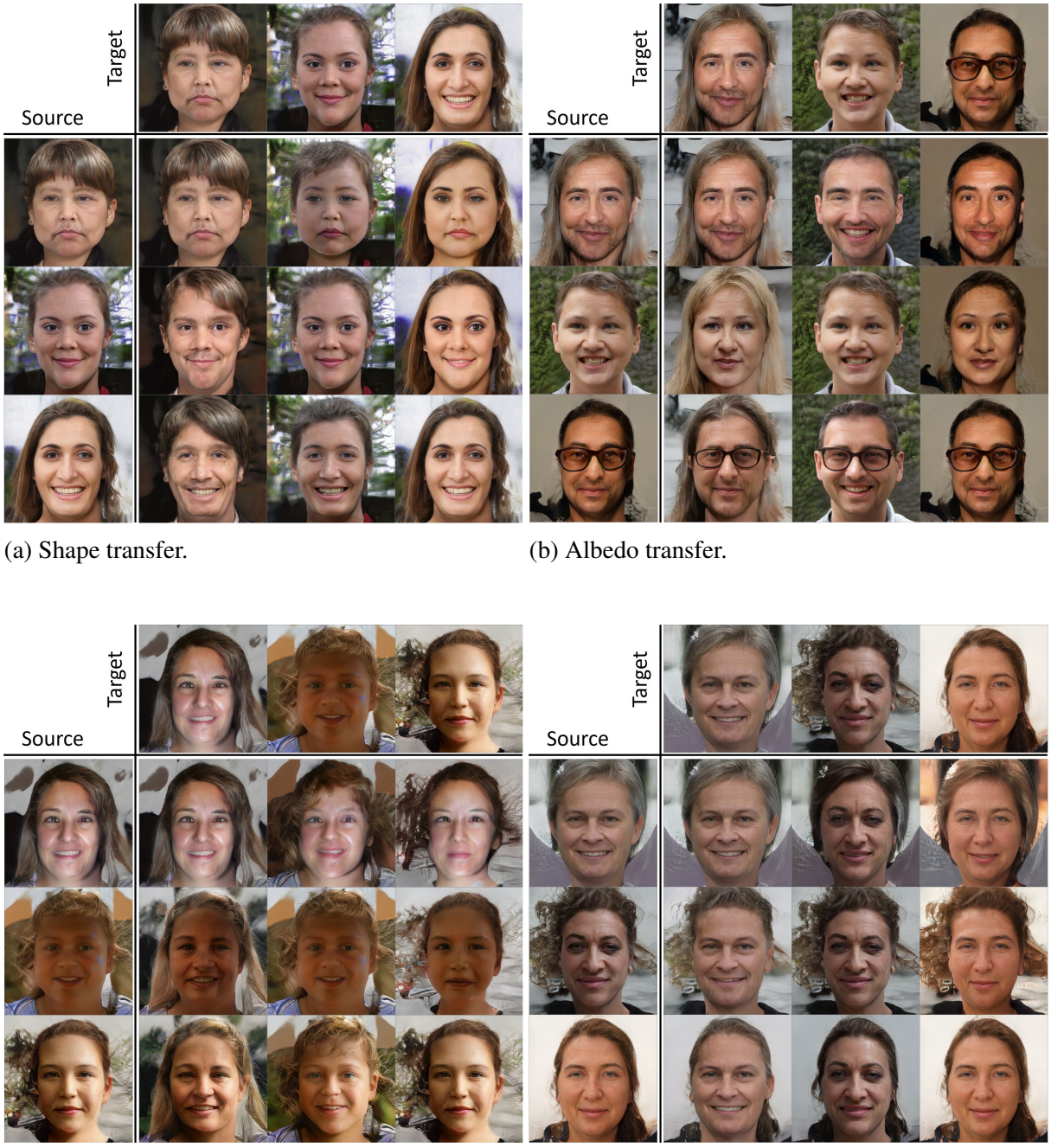
Shape, Albedo, Lighting and Hairstyle Manipulation: To demonstrate the disentanglement capability of our model, we manipulate the shape, albedo, lighting, and hair and background of generated faces and show their variations (see Fig. 1.4). For a face image generated using some latent code $\mathbf{z}$, we modify attributes of the image by independently resampling one or more of $\mathbf{z}_\alpha$, $\mathbf{z}_\tau$, $\mathbf{z}_\gamma$ and $\mathbf{z}_{\text{hairbg}}$ from the latent space and replacing the original values by the resampled values for the selected portions of $\mathbf{z}$. Then we use the modified latent code to generate a modified image. While MOST-GAN's shape parameters $\mathbf{z}_\alpha$ correspond to both identity and expression variations, many individual dimensions of $\mathbf{z}_\alpha$ correspond more with either identity or expression. By altering the values in these dimensions, we can change the face shape to selectively focus on either identity-related or expression-related shape changes, as shown in Fig. 1.4. Altering the albedo code results in changes to properties such as lip color, skin tone, facial hair, and eyebrow density, while leaving the face shape virtually unchanged. Similarly, varying the illumination and hair/background latent codes only affect those factors, while maintaining the face's shape and albedo.

Illumination Editing using Spherical Harmonics: Since MOST-GAN's illumination code is based on the spherical harmonics coefficients [Ramamoorthi and Hanrahan, 2001], we can perform controlled manipulation of illumination by directly setting the values of the spherical harmonics coefficients, then using Eq. (4.8) to map these values into the space of $\boldsymbol{z}_\gamma$. We traverse through the first two spherical harmonics bases for each channel and show the illumination variations in Fig. 4.3. Traversing through the first basis results in global illumination change, while traversing through the second basis results in the illumination direction changing from right to left. Notice that as the magnitude and direction of light changes, it affects not only the face but also the hair and background. This is in contrast to 3DMM-based methods like MOST-GAN, which apply illumination only to the face region. As a result, illumination editing using our method is more natural than that of 3DMM-based approaches.

To further demonstrate the correspondence between CoLa-SDF's illumination latent code and the spherical harmonics coefficients [Ramamoorthi and Hanrahan, 2001], we show controlled il-

Figure 4.3 Illumination editing using spherical harmonics. For three randomly generated faces, we can alter the lighting by directly modifying the spherical harmonics coefficients. Varying the first spherical harmonics coefficient (left) controls the level of global (ambient) illumination, while the second coefficient (right) controls the illumination's horizontal directionality.

Figure 4.4 Directional rotation of illumination.



(a) Shape transfer.

(b) Albedo transfer.

(c) Lighting transfer.

(d) Hair transfer.

Figure 4.5 Transfering physical attributes from source to target through the latent code.

lumination manipulation in Fig. 4.4. Starting from an initial illumination setting (shown in the left column), we project it to the spherical harmonics space using Eq. (4.8) and rotate the lighting around the camera axis in increments of $\pi/5$ radians $(36°)$. The results demonstrate that CoLa-SDF can perform any desired illumination editing.

4.3.4.1 Attribute Transfer

To further demonstrate the attribute disentanglement of our method, we transfer attributes such as shape, albedo, lighting, and hair and background from a source image (left column) to a target image (top row), as illustrated in Fig. 4.5.

Shape Transfer (Fig. 4.5a): Our method can transfer extreme identity- and expression-related shape variations from the source image to the target image, while keeping other physical attributes intact. These changes include width and height of the face, roundness of the face (row 1), sharpness of the jawline (row 4), as well as expression changes such as frowning (row 1) and smiling.

Albedo Transfer (Fig. 4.5b): Our model can transfer attributes such as skin tone, thickness of eyebrows (rows 3 and 4), and lip color. Interestingly, our model can also transfer eyeglasses, which are external to the face and hence not accounted for by any 3DMM model. In addition, we were surprised to observe that hair color is affected by the albedo code in addition to the hair and background code.

Illumination Transfer (Fig. 4.5c): While skin color is a property of facial albedo, we note that the illumination code can change the tone, hue and brightness of the overall image, including hair and background.

Hair and Background Transfer (Fig. 4.5d): Notice that transferring the hair and background does not change the identity or other attributes of the face. In this figure, we again observe that while the hair/background code determines the hair geometry/hairstyle, its color is also partly controlled by the albedo code.

4.3.4.2 Identity Consistency across Unrelated Attributes

In this section, we analyze the effect on the identity of the generated face of changing identity-related attributes such as shape and albedo versus non-identity-related attributes such as pose, illumination, and hair and background. We randomly generated 1000 face images from our model and edited their viewpoint, illumination, hair/background, shape, and albedo by resampling their latent codes from the corresponding normal distributions. We extract the identity features from the

Face identity match (% of samples with unchanged identity) between original and edited images.					
View	**Illumination**	**Hair/backround**	**Shape**	**Albedo**	**Shape + Albedo**
97.7	99.7	98.2	75.2	65.7	4.7

Table 4.2 Evaluation of face identity consistency as measured by ArcFace [Deng et al., 2019] after resampling non-identity-related latents (view, illumination and hair/background), and identity-related latents (shape, albedo).

(a) Ours (b) Without face loss $\mathcal{L}_{\text{face}}$

Figure 4.6 Without face loss $\mathcal{L}_{\text{face}}$, the hair/background latent code does not get fully disentangled from the face. This leads to changes in the face region with the hair/background latent code as can be seen in the examples to the right.

original and the edited images using the state-of-the-art face-recognition model ArcFace [Deng et al., 2019], and measure the identity match between the original and edited faces (using ArcFace threshold of 70°). The results, in Tab. 4.2, show that as desired, changes in viewpoint, illumination, and hair and background have minimal impact on the generated face's identity. In contrast, changing shape and albedo individually cause partial but not complete identity alterations (this corresponds well with human perception of identity changes in Figs. 4.5a and 4.5b), while simultaneously changing both shape and albedo codes results in a clear change of identity. This demonstrates that our method has successfully disentangled the identity-related attributes of face from its non-identity-related attributes.

4.3.5 Ablation Studies

The development of CoLa-SDF involved a number of important design choices. To show the effects of some of these choices, these ablation studies demonstrate how various omissions from or additions to CoLa-SDF detract from its overall performance.

4.3.5.1 Without face consistency loss

CoLa-SDF-NoFaceLoss: As described in Sec. 4.2.4.2, we enforce hair/background disentanglement through a combination of the hair/background consistency loss $\mathcal{L}_{\text{hairbg}}$ and the face consistency loss $\mathcal{L}_{\text{face}}$. The hair/background consistency loss, $\mathcal{L}_{\text{hairbg}}$, ensures that when we keep the hair/background code $\mathbf{z}_{\text{hairbg}}$ the same but change the other latent codes, the hair/background regions in the image will change as little as possible. Similarly, $\mathcal{L}_{\text{face}}$ ensures that when we change $\mathbf{z}_{\text{hairbg}}$ but keep the other latent codes the same, the face region will change as little as possible.

To study the importance of the face consistency loss, we train a model without $\mathcal{L}_{\text{face}}$ loss and evaluate it in terms of its hair/background disentanglement. We call this variant CoLa-SDF-NoFaceLoss. Specifically, we perform interpolation between two hair/background codes while keeping all the other latent codes the same. If the model has well disentangled hair/background from the face region, changing $\mathbf{z}_{\text{hairbg}}$ should not affect the face. We show the comparison between our model and CoLa-SDF-NoFaceLoss in Fig. 4.6. Note that, with CoLa-SDF-NoFaceLoss, changing $\mathbf{z}_{\text{hairbg}}$ changes facial-hair in the first row, and causes shape changes in the second row. On the other hand, with our model, changing $\mathbf{z}_{\text{hairbg}}$ does not any cause noticeable changes in the face regions.

4.3.5.2 Independent mapping of each attribute:

CoLa-SDF-SeparateMappers: In CoLa-SDF, the five latent codes $\mathbf{z}_{\alpha}$, $\mathbf{z}_{\tau}$, $\mathbf{z}_{\gamma}$, $\mathbf{z}_{\text{hairbg}}$, and $\mathbf{z}_{\text{rest}}$ all feed into the same Renderer Mapping Network, which outputs a combined style code $\mathbf{w}$, as shown in the top left of Fig. 4.1. For this ablation study, we replace the single volume renderer mapping network with five separate renderer mapping networks, one for each of shape $\boldsymbol{\alpha}$, albedo $\boldsymbol{\tau}$, illumination $\boldsymbol{\gamma}$, hair/background $\mathbf{z}_{\text{hairbg}}$, and $\mathbf{z}_{\text{rest}}$. We sample the shape parameters from $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}_{\alpha}, \boldsymbol{\Sigma}_{\alpha})$, where $\boldsymbol{\mu}_{\alpha}$ and $\boldsymbol{\Sigma}_{\alpha}$ are the data mean and covariance obtained from MOST-GAN encodings of the FFHQ dataset [Karras et al., 2020] images. We used the same method to sample $\boldsymbol{\tau}$ and $\boldsymbol{\gamma}$. For sampling $\mathbf{z}_{\text{hairbg}}$ and $\mathbf{z}_{\text{rest}}$, we use the standard normal distribution $\mathcal{N}(0, 1)$. The individual mappers have similar architecture as the original combined renderer mapping network, but with different input and output dimensions. The input dimensions for shape, albedo, illumination, hairbg, and rest are

	Ablation Variants			
	Separate Mappers	With Perceptual Consistency	With Photometric Consistency	Ours
FID (↓)	25.85	23.04	21.38	**19.4**

Table 4.3 FID evaluations at 256x256 resolution. CoLa-SDF with Separate Mappers performs the worst, while enforcing photometric or perceptual consistency losses also harm the FID scores. Our proposed method scores the best FID scores while maintaining latent space disentanglement.

150, 200, 27, 64, 64, respectively. Their output dimensions are 37, 64, 27, 64, 64, respectively, to match the latent code factorization of CoLa-SDF. We concatenate the shape, albedo, illumination, hairbg, and rest style-codes obtained from these independent mappers to form the combined style code, $\mathbf{w}$, which is then passed through the rest of the algorithm exactly as in CoLa-SDF. Note that we adopt separate mappers only during the volume renderer phase; the generator mapping network remains a single network as in CoLa-SDF. This variant, though, results in a loss of image quality and diversity, as evaluated in terms of FID [Heusel et al., 2017] (see Tab. 4.3).

4.3.5.3 Add low-res to high-res consistency loss

CoLa-SDF-Photometric: In this variant, we adopt a photometric consistency loss $\mathcal{L}_{\text{photocons}}$ to enforce consistency between the high-resolution image obtained from the styled generator (after downsampling it to the low-resolution scale), and the low-resolution image obtained from the volume renderer:

$$\mathcal{L}_{\text{photocons}} = ||\text{down}(\mathbf{I}_{gen},\ \text{size}(\mathbf{I}_{vol}) - \mathbf{I}_{vol}||_2^2, \tag{4.15}$$

where $\text{down}(\mathbf{x}, (h, w))$ downsamples image $\mathbf{x}$ to height h and width w using bilinear interpolation. This acts as an additional loss to ensure that the disentanglement of physical attributes in the volume renderer reflects well in the styled-generator too.

CoLa-SDF-Perceptual: This variant is similar to CoLa-SDF-Photometric, except that we replace the photometric consistency loss with a perceptual consistency loss [Zhang et al., 2018b]:

$$\mathcal{L}_{\text{vggcons}} = ||\phi(\text{down}(\mathbf{I}_{gen},\ \text{size}(\mathbf{I}_{vol})) - \phi(\mathbf{I}_{vol})||_2^2, \tag{4.16}$$

where ϕ is the VGGFace [Parkhi et al., 2015] model.

We found that, both these variants lead to higher FID metrics as reported in Tab. 4.3, which is an indicator of low image quality and diversity.

4.3.6 Limitations

CoLa-SDF may sometime generate artifacts during hair/background editing as shown in Fig. 4.7. We believe this is due to the model's incapability to differentiate between hair and cap, and ending up interpolating between them. In addition, as observed in Figs. 4.5b and 4.5d, CoLa-SDF has learned to model hair geometry through the hair/background code, but hair color is controlled by a combination of the albedo and hair/background codes. This may be due to a correlation between hair-color and texture in the training dataset.

Figure 4.7 Spurious artifacts during hairstyle editing.

4.4 Conclusion

We propose a new method called CoLa-SDF that combines the disentangled controllability of nonlinear 3DMM approaches with the high fidelity of implicit 3D GANs for generating 3D faces and rendering them to images. Building upon the architecture of StyleSDF, we enforce the latent space to match the physical parameters of the nonlinear 3D morphable model MOST-GAN, as well as disentangling control of hair and background, a feat we believe is a first of its kind. We demonstrate high-fidelity image synthesis and subsequent 3D manipulation with full control over the disentangled latent parameters. Overall, the proposed model presents a promising solution for generating high-quality 3D faces with controllable properties, which can have practical applications in many areas including AR/VR, dataset synthesis and augmentation, media, and avatar creation.

CHAPTER 5

DIVERSE3DFACE: TOWARDS ROBUST AND DIVERSITY-PROMOTING 3D FACE RECONSTRUCTION FROM SINGLE-VIEW IMAGES

Dey, R. and Boddeti, V. N. Generating Diverse 3D Reconstructions from a Single Occluded Face Image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1547-1557, 2022.

Single image-based 3D face reconstruction has improved significantly in recent years [Zollhöfer et al., 2018, Egger et al., 2020]. This includes advances in statistical models [Blanz and Vetter, 1999, Paysan et al., 2009, Li et al., 2017a, Ploumpis et al., 2020] as well as neural network-based models [Tewari et al., 2017, Tran and Liu, 2019, Sengupta et al., 2018, Wu et al., 2020, Feng et al., 2021, Gecer et al., 2019, Tran et al., 2019, Wei et al., 2019, Tuan Tran et al., 2017]. However, facial occlusions remain a significant challenge to this task. In-the-wild face images often come with several forms of occlusions and unless dealt with explicitly, often lead to erroneous 3D reconstruction in terms of shape, expression, pose, *etc*. [Egger et al., 2020, Egger et al., 2018, Tuán Trán et al., 2018].

3D reconstruction of partially occluded faces presents two main challenges. First, 3D reconstruction models need to selectively use features from the visible regions while ignoring those from the occluded parts. Failure to do so, either implicitly or explicitly, will lead to poor 3D reconstructions with an incorrect pose, expression, or both. Second, there could be a distribution of 3D reconstructions that are consistent with the visible parts in the image yet diverse on the occluded parts. Failure to account for all such modes limits the utility of 3D reconstruction models. Addressing these two challenges is the primary goal of this paper.

Existing 3D face reconstruction solutions, however, are ill-equipped to overcome both of these challenges simultaneously. From a **reconstruction perspective**, a majority of the approaches that reconstruct 3D faces from a single image restrict themselves to fully-visible face images. And, even those that explicitly account for facial occlusions [Tuán Trán et al., 2018, Egger et al., 2018],

do so only in a holistic manner using a global model that implicitly uses features from the occluded regions as well. This form of global model-based fitting can introduce errors (see Fig. 1.5) in the pose and expression of the 3D reconstruction, especially when large portions of the face are occluded. From a **diversity perspective**, existing approaches are, by design, limited to only generating a single plausible 3D reconstruction. However, in many practical applications, for a single occluded face image, it is desirable to generate multiple reconstructions that are consistent on the visible parts of the face, while spanning a diverse yet realistic set of reconstructions on the occluded parts (see Fig. 1.5). While the concept of generating diverse solutions has been explored in other contexts such as image generation [Elfeki et al., 2019], image completion [Zheng et al., 2019b], super-resolution [Bahat and Michaeli, 2020] and trajectory forecasting [Yuan and Kitani, 2019], they have not been explored for monocular 3D face reconstruction of occluded faces.

In this work, we propose Diverse3DFace [Dey and Boddeti, 2022b], which is designed to simultaneously yield a diverse, yet plausible, set of 3D reconstructions from a single occluded face image. Diverse3DFace consists of three modules: a global + local shape fitting process, a graph neural network based variational autoencoder (Mesh-VAE), and a Determinantal Point Process (DPP) [Kulesza and Taskar, 2012] based iterative optimization procedure. The global + local shape fitting process affords robustness against large occlusions by decoupling shape fitting on the visible regions from that of the occluded regions. The Mesh-VAE enables to learn a distribution over a compact latent space over the different factors of variation in the 3D shapes of faces. And, the DPP-based iterative optimization procedure enables us to sample from the latent space of the Mesh-VAE and optimize them to generate a diverse set of reconstructions spanning the different modes of the latent space. Our specific contributions in this paper are:

– We propose Diverse3DFace, a simple yet effective diversity promoting 3D face reconstruction approach that generates multiple plausible 3D reconstructions corresponding to a single occluded face image.

– For robustness to occlusions, we propose a global+local PCA model-based shape fitting that disentangles the fitting on each facial component from the others. The models are learned from

a dataset of FLAME [Li et al., 2017a] registered 3D meshes. During inference, the local perturbations on various facial components are added on top of a coarse global fit to generate the final detailed fitting.

– We employ a DPP [Kulesza and Taskar, 2012] based diversity loss in the context of generating diverse 3D reconstructions of faces. We define the quality and similarity terms in the DPP kernel to maximize diversity while remaining in the space of realistic 3D head shapes.

– We conduct extensive qualitative and quantitative experiments to show the efficacy of the proposed approach in generating 3D reconstructions that are faithful to the visible face while simultaneously capturing multiple diverse modes on the occluded parts. The solution from Diverse3DFace that is closest to the ground truth is on average 30-50% better than the unique solutions of the baselines [Feng et al., 2021, Li et al., 2017a] in terms of per-vertex ℓ_2-error.

5.1 Preliminaries

Determinantal Point Processes: Determinantal Point Processes (DPPs) originated in quantum physics to model the negative correlations between the quantum states of fermions [Macchi, 1975]. DPPs were first introduced in machine learning by Kulesza and Taskar [Kulesza and Taskar, 2012] as a probabilistic model of repulsion between points. A point process over a ground set $\mathcal{Y}$ describes the probability of all its $2^{\mathcal{Y}}$ subsets. A point process is determinantal when the probability of choosing a random subset $Y \subseteq \mathcal{Y}$ is given by the determinant of the sub-kernel matrix $\mathbf{L}_Y$ indexed by the elements of Y, *i.e.*, $P(Y \subseteq \mathcal{Y}) = det(\mathbf{L}_Y)$. Given a data matrix $B \in \mathbb{R}^{D \times N}$, we can compute the kernel as the Gram matrix $\mathbf{L} = B^T B$. In this case, the determinant of the sub-kernel matrix $det(\mathbf{L}_Y)$ is related to the volume spanned by the elements of B. Thus, conceptually, DPP assigns a higher probability to a subset whose elements tend to be orthogonal (diverse) to each other, thus spanning a larger volume.

5.2 Approach

Reconstructing diverse 3D shapes in a single stage, using only a global model, is sub-optimal due to multiple reasons, as we show in our experiments (Sec. 5.3.1). First, fitting a global model to

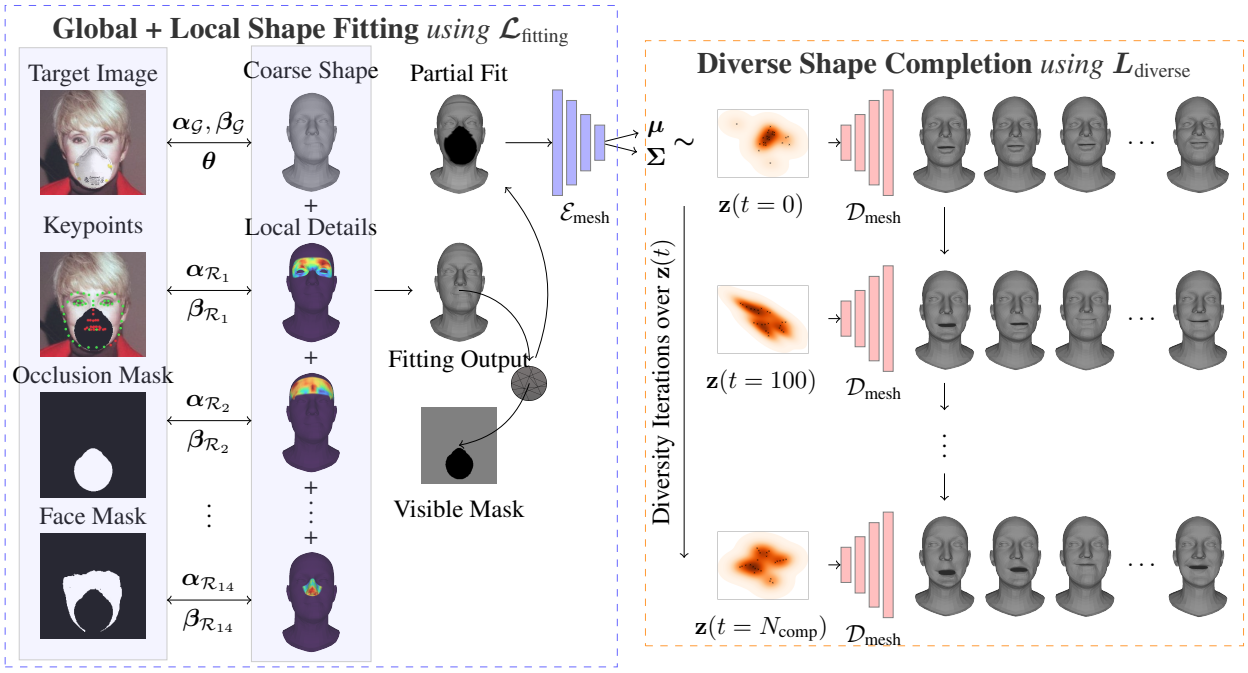


Figure 5.1 Overview: As input, we need the target image, the occlusion mask, facial landmarks, and optionally a face mask. We use the HRNET model [Wang et al., 2020] to obtain both the landmark locations and their confidence values, which we use to estimate the occlusion labels. Given these input, we first fit our proposed *global + local blendshape model* to obtain the coarse and local fittings as outlined in Algorithm 5.1, which we then add together to obtain the final fitting. We re-project the fitted shape onto the visible mask to obtain a partial fit, zeroed out on the occluded regions. We map the partial fit onto a latent space using the *Mesh-VAE* encoder $\mathcal{E}_{\text{mesh}}$ and sample N latent vectors $\mathbf{z}$. We then iteratively optimize the $\mathbf{z}$'s to capture diverse modes with respect to the occluded regions while remaining consistent with the visible regions as outlined in Algorithm 5.2 to obtain the final set of 3D reconstructions.

a few visible sub-regions requires striking a careful trade-off between robustness and local fidelity which is challenging to achieve. Second, diversification of the occluded regions will inadvertently affect the quality of fitting on the visible regions, and vice-versa. Given these observations, we propose a three-step approach to generate diverse, yet realistic 3D reconstructions from an occluded face image. In step 1, we use an ensemble of disentangled global+local shape models to perform robust 3D reconstruction with respect to the visible parts of the face. In step 2, we employ a VAE to map the partial fit to a latent space from which multiple reconstructions can be drawn. Finally, in step 3 we iteratively optimize the latent embeddings to promote realistic geometric diversity on the occluded face regions while maintaining fidelity to the visible ones. We now describe our complete algorithm along with its different components.

5.2.1 Global + Local Shape Model

A robust partial 3D reconstruction that accurately fits the visible parts of the face is a prerequisite for generating diverse solutions. Existing approaches of occlusion-robust 3D reconstruction typically employ a global model to fit or regress based on the visible regions [Egger et al., 2018, Tuán Trán et al., 2018]. Because of the *global* nature of such models, errors in occlusion segmentation affect the quality of 3D reconstruction [Saito et al., 2016], even on the visible parts (see Fig. 5.3). Typically, strong regularization is employed to mitigate such effects. However, while heavier regularization leads to more robustness against occlusions, it comes at the cost of sub-optimal fitting. This observation, along with the successful application of localized deformation components in computer graphics [Neumann et al., 2013, Schwartz et al., 2020], motivated us to adopt an ensemble of global + local models as an effective approach to generate robust 3D reconstructions with respect to the visible parts. Note that, in this stage of our solution, we are not concerned about the reconstruction quality in the occluded regions. We now describe the details of our proposed global+local 3D head model.

Our global+local shape model is based on the FLAME mesh topology [Li et al., 2017a]. We use the FLAME registered D3DFACS [Cosker et al., 2011] and CoMA [Ranjan et al., 2018] datasets to compute the local PCA models. The FLAME [Li et al., 2017a] model comes with vertex masks corresponding to 14 parts on the human head. We trained individual PCA models corresponding to each of these parts to account for local variations. To do so, we first take FLAME-registered meshes and fit the full FLAME model [Li et al., 2017a] to these by optimizing the following fitting loss:

$$\mathcal{L}_{\text{gtfit}} = \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta}} ||S_{\text{gt}} - \tilde{S}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})||, \tag{5.1}$$

Here S_{gt} is the ground-truth shape and $\tilde{S}(\boldsymbol{\alpha},\boldsymbol{\theta},\boldsymbol{\beta})$ is obtained using Eqs. (2.1) and (2.2). We then *unpose* both the ground-truth and the fitted shapes by removing the variations due to pose $\boldsymbol{\theta}$ as described in [Li et al., 2017a] and obtain S^0_{gt} and $\tilde{S}(\boldsymbol{\alpha}, 0, \boldsymbol{\beta})$, respectively. The full FLAME model consists of $|\boldsymbol{\alpha}|= 300$ shapes and $|\boldsymbol{\beta}|= 100$ expression bases to account for complete global variations. From this, we retain the top N_S shape and N_E expression bases (based on eigenvalues)

and discard the rest to compute shape residuals $\tilde{S}_{\text{res}} = S^0_{\text{gt}} - \tilde{S}_{\text{coarse}}$, where

$$\tilde{S}_{\text{coarse}} = \bar{S} + \sum_{n=1}^{N_S} \boldsymbol{\alpha}_n \mathcal{S}_n + \sum_{n=1}^{N_E} \boldsymbol{\beta}_n \mathcal{E}_n \tag{5.2}$$

We then compute the region-wise shape and expression PCA models $(\mathcal{S}_{\mathcal{R}_i}, \mathcal{E}_{\mathcal{R}_i})$ using the region-wise residuals $M_{\mathcal{R}_i} \odot \tilde{S}_{\text{res}}$ (here $M_{\mathcal{R}_i}$ is the vertex-mask for the i-th region). For computing the shape bases, we set $N_S = 10$ and $N_E = 100$ (removing all expression variations); while for the expression bases, we set $N_E = 10$ and $N_S = 300$ (removing all identity variations). The global + local model can then be represented as,

$$S(\boldsymbol{\alpha}_{\mathcal{G}}, \boldsymbol{\alpha}_{\mathcal{R}}, \boldsymbol{\beta}_{\mathcal{G}}, \boldsymbol{\theta}, \boldsymbol{\beta}_{\mathcal{R}}) = S_{\mathcal{G}}(\boldsymbol{\alpha}_{\mathcal{G}}, \boldsymbol{\beta}_{\mathcal{G}}, \boldsymbol{\theta}) + S_{\mathcal{R}}(\boldsymbol{\alpha}_{\mathcal{R}}, \boldsymbol{\beta}_{\mathcal{R}}), \tag{5.3}$$

where $S_{\mathcal{G}}(\boldsymbol{\alpha}_{\mathcal{G}}, \boldsymbol{\beta}_{\mathcal{G}}, \boldsymbol{\theta})$ is the coarse global shape given by the top N_S shape and N_E expression global bases, along with the pose blendshapes $\mathcal{P}$ (Eq. (2.2)); and $S_{\mathcal{R}}(\boldsymbol{\alpha}_{\mathcal{R}}, \boldsymbol{\beta}_{\mathcal{R}})$ represent the sum of all local variations and is given by,

$$S_{\mathcal{R}}(\boldsymbol{\alpha}_{\mathcal{R}}, \boldsymbol{\beta}_{\mathcal{R}}) = \sum_{\mathcal{R}_i, i=1}^{14} (\mathcal{S}_{\mathcal{R}_i} \boldsymbol{\alpha}_{\mathcal{R}_i} + \mathcal{E}_{\mathcal{R}_i} \boldsymbol{\beta}_{\mathcal{R}_i}) \tag{5.4}$$

5.2.2 Shape Completion using Mesh-VAE

We use the global+local model to fit robust 3D reconstruction on the visible parts of the occluded face. But this does not ensure robust and consistent reconstruction on the occluded parts since the local PCA models have noisy (occluded) or no data to fit to. To address this drawback and to enable the generation of a distribution of plausible 3D reconstructions rather than a singular solution, which is one of our primary goals, we adopt a mesh-based VAE (dubbed *Mesh-VAE*) as our shape completion model.

We assume that human head meshes can be mapped onto a continuous and regularized low-dimensional latent space $\mathcal{Z}$. Then, given a masked (partial) 3D mesh S_m, the Mesh-VAE learns the conditional likelihood of mesh completions S_c and the corresponding latent embeddings $\mathbf{z}$:

$$p(S_c, \mathbf{z}|S_m) = p(\mathbf{z}|S_m) p(S_c|\mathbf{z}, S_m), \tag{5.5}$$

5.2.3 DPP Driven Shape Diversification

Even though the Mesh-VAE can sample multiple shape completions from $p(S_c|\mathbf{z}, S_m)$, in practice, the generated samples from a VAE are not guaranteed to cover all the modes [Yuan and Kitani, 2019] (see Sec. 5.3.1). To enforce diversity, we formulate a DPP on shape completions and develop a diversity loss to optimize their latent embeddings.

We adopt the quality-diversity based formulation of the DPP kernel $\mathbf{L}$ [Kulesza and Taskar, 2012], which seeks to balance the quality of samples with their diversity. Specifically, for elements i, j in a set, its kernel entry is given by $L_{i,j} = q_i S_{i,j} q_j$, where q_i denotes the quality of element i, and $S_{i,j}$ represents the similarity between i and j. Maximizing the determinant of such a kernel matrix implies maximizing the quality of each sample while minimizing the similarity between distinct samples. For two shape completions $S_{c,i}$ and $S_{c,j}$, we define the similarity as

$$S_{i,j} = \exp\left(-\frac{k}{\text{median}_{i,j}(\text{dist}_{i,j})}\text{dist}_{i,j}\right), \tag{5.6}$$

where $\text{dist}_{i,j} = ||S_{c,i} - S_{c,j}||_2$ is the ℓ_2 distance between the i-th and j-th shape completions and k is a scaling factor. To ensure that the completed samples look realistic, we relate the quality of a sample with the probability of its latent embedding $\mathbf{z}_i$ lying within 3σ of the prior $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as:

$$q_i = \exp(-\max(0, \mathbf{z}_i^T \mathbf{z}_i - 3\sqrt{d})), \tag{5.7}$$

where d is the dimensionality of $\mathbf{z}_i$. For numerical stability [Yuan and Kitani, 2019], we adopt expected cardinality of $\mathbf{L}$ as the DPP loss:

$$\mathcal{L}_{\text{dpp}} = -\text{tr}\left(I - (\mathbf{L} + I)^{-1}\right), \tag{5.8}$$

where I is the identity matrix, and $\text{tr}(.)$ represents the trace of a matrix.

5.2.4 Inference

Given an occluded face image $\mathbf{I}_m$, our goal is to generate a distribution of plausible 3D reconstructions $S_{c,1}, ..., S_{c,M}$. We do this in three steps which we describe below:

Step 1 *Partial Shape Fitting:* In this stage, we first fit our global + local PCA model on the visible parts of the face image $\mathbf{I}_m$ to obtain a partial reconstruction S_m. We employ the following fitting

loss:

$$\mathcal{L}_{\text{fitting}} = \lambda_1^f \mathcal{L}_{\text{lmk}} + \lambda_2^f \mathcal{L}_{\text{pho}} + \lambda_3^f \mathcal{L}_{\text{reg}}, \tag{5.9}$$

where $\mathcal{L}_{\text{lmk}}$ is the landmark loss, $\mathcal{L}_{\text{pho}}$ is the photometric loss and $\mathcal{L}_{\text{reg}}$ applies ℓ_2-regularization over the model parameters. We use an off-the-shelf landmark detector HRNET [Wang et al., 2020] to detect 68 landmarks on the face along with their confidence values. We mark those landmarks as visible whose confidence exceeds a threshold ϵ (set to 0.2) and apply the landmark loss on those points. To add local details, we apply an ℓ_1-based photometric loss between the input image and the rendered image $\mathbf{I}_{\text{ren}}$ on the visible regions, where $\mathbf{I}_{\text{ren}} = \mathcal{R}(S_m, B_\tau(\boldsymbol{\tau}, \mathcal{A}), \boldsymbol{\gamma}, \mathbf{c})$, $\boldsymbol{\tau}$ are the fitted albedo parameters, $\mathcal{A}$ are the orthonormal albedo bases from [Li et al., 2017a], $B_\tau(\boldsymbol{\tau}, \mathcal{A}) = \mathcal{A}\boldsymbol{\tau}$ and $\mathbf{c}$ is the estimated camera parameters. We restrict the photometric loss to the visible face region using the face mask M_f and the occlusion mask $\mathbf{M}_o$:

$$\mathcal{L}_{\text{pho}} = ||(\mathbf{I}_m - \mathbf{I}_{\text{ren}}) \odot M_f \odot (\mathbf{1} - \mathbf{M}_o)||_1 \tag{5.10}$$

Step 2 We use the encoder to map the partial fit S_m to a latent distribution from which we sample the latent embeddings $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$, where $\boldsymbol{\mu}, \boldsymbol{\sigma} = \mathcal{E}_{\text{mesh}}(S_m)$.

Step 3 *Diversity Promoting Shape Completion:* In this stage, we perform a diversity promoting iterative shape completion routine, which forces the latent embeddings towards diverse modes w.r.t the occluded regions while remaining faithful to the visible regions. At each iteration, we obtain a distribution of shape completions using the decoder $S_{c,j} = \mathcal{D}_{\text{mesh}}(\mathbf{z}_j), \forall j = 1...M$, and update the $\mathbf{z}$'s to minimize a diversity loss:

$$\mathcal{L}_{\text{diversity}} = \lambda_1 \mathcal{L}_S + \lambda_2 \mathcal{L}_{\text{pho}} + \lambda_3 \mathcal{L}_{\text{dpp}} \tag{5.11}$$

Here $\mathcal{L}_S$ is the shape consistency loss defined as the ℓ_1-norm between the $S_{c,j}$'s and S_m applied on the visible vertices, $\mathcal{L}_{\text{pho}}$ is the photometric loss (Eq. (5.10)) and $\mathcal{L}_{\text{dpp}}$ is the DPP loss (Eq. (5.8)). The loss coefficients are set to have similar magnitude for all the loss components.

We outline the full steps for partial shape fitting and diversification in Algorithm 5.1 and Algorithm 5.2, respectively.

Algorithm 5.1 Shape Fitting on the Visible Face Regions

Input: Image $\mathbf{I}_m$, Occlusion mask $\mathbf{M}_o$, Face mask $\mathbf{M}_f$, Global models $\mathcal{S}, \mathcal{E}, \mathcal{P}$, Local models $\mathcal{S}_{\mathcal{R}_i}$, $\mathcal{E}_{\mathcal{R}_i}$ for $i = 1$ to 14, Albedo model $\mathcal{A}$, Landmarks detector $\mathcal{H}$
Parameters: $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\tau}, \mathbf{c}, \boldsymbol{\alpha}_{\mathcal{R}_i}, \boldsymbol{\beta}_{\mathcal{R}_i}$ for $i = 1$ to 14
Hyperparameters: $\epsilon = 0.1, n_{\text{iter}}, \lambda_1^f, \lambda_2^f, \lambda_3^f, \eta$
Output: Partially fitted shape S_m

Detect landmarks from image $L_I, L_{\text{conf}} \leftarrow \mathcal{H}(\mathbf{I}_m)$
Set $L_{\text{valid}} \leftarrow 1$ when $L_{\text{conf}} > \epsilon$ else 0
for $j = 1$ to n_{iter} **do**
  Obtain S_m using Eqs. (2.1), (2.2), (5.3) and (5.4)
  Select 68 landmarks from shape $L_S \leftarrow M_{\text{lmk}}(S)$
  Obtain rendered image $\mathbf{I}_{\text{ren}} \leftarrow \mathcal{R}(S_m, B_\tau(\boldsymbol{\tau}, \mathcal{A}), \boldsymbol{\gamma}, \mathbf{c})$
  $\mathcal{L}_{\text{lmk}}^f \leftarrow ||(L_S - L_I) \odot L_{\text{valid}}||_1$
  $\mathcal{L}_{\text{pho}}^f \leftarrow ||(\mathbf{I}_m - \mathbf{I}_{ren}) \odot \mathbf{M}_f \odot (\mathbf{1} - \mathbf{M}_o)||_1$
  $\mathcal{L}_{\text{reg}}^f \leftarrow \ell_2$ regularization loss over all parameters
  $\mathcal{L}_{\text{fitting}} = \lambda_1^f \mathcal{L}_{\text{lmk}}^f + \lambda_2^f \mathcal{L}_{\text{pho}}^f + \lambda_3^f \mathcal{L}_{\text{reg}}^f$
  Update $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_w \mathcal{L}_{\text{fitting}}$ for $\mathbf{w} \in \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\tau}, \boldsymbol{\gamma}, \mathbf{c}, \boldsymbol{\alpha}^{\mathcal{R}_i}, \boldsymbol{\beta}^{\mathcal{R}_i}$ for $i = 1$ to 14
end for

Algorithm 5.2 Diverse Shape Completions

Input: Mesh-VAE Encoder $\mathcal{E}_{\text{mesh}}$ and Decoder $\mathcal{D}_{\text{mesh}}$
Input from Algorithm 5.1: $\mathbf{I}_m$, $\mathbf{M}_o$, $\mathbf{M}_f$, L_I, L_{valid}, $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\tau}$, $\mathbf{c}$
Hyperparameters: $n_{\text{comp}}, \lambda_1, \lambda_2, \lambda_3, \eta$
Output: M Shape completions $\{S_{c,j=1:M}\}$

Sample the vertex mask $\mathbf{M}_o^v$ by projecting S onto $\mathbf{M}_o$
Obtain latent parameters $\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow \mathcal{E}_{\text{mesh}}(S_m \odot \mathbf{M}_o^v)$
Sample M latent vectors $\mathbf{z}_1, ..., \mathbf{z}_M \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$
for $i = 1$ to n_{comp} **do**
  Obtain $S_{c,j} \leftarrow \mathcal{D}_{\text{mesh}}(\mathbf{z}_j)$ for $j = 1...M$
  Obtain $\mathbf{I}_{\text{ren},j} \leftarrow \mathcal{R}(S_{c,j}, B_\tau(\boldsymbol{\tau}, \mathcal{A}), \boldsymbol{\gamma}, \mathbf{c})$ for $j = 1...M$
  $\mathcal{L}_S \leftarrow \sum_{j=1}^M ||(S_{c,j} - S_m) \odot (\mathbf{1} - \mathbf{M}_o^v)||_1$
  $\mathcal{L}_{\text{pho}} \leftarrow \sum_{j=1}^M ||(\mathbf{I}_m - \mathbf{I}_{ren,j}) \odot \mathbf{M}_f \odot (\mathbf{1} - \mathbf{M}_o)||_1$
  $\mathcal{L}_{\text{dpp}} \leftarrow \mathcal{L}_{\text{dpp}}(S_c^{j=1:M} \odot \mathbf{M}_o^v)$ using Eq. (5.8)
  $\mathcal{L}_{\text{diversity}} = \lambda_1 \mathcal{L}_S + \lambda_2 \mathcal{L}_{\text{pho}} + \lambda_3 \mathcal{L}_{\text{dpp}}$
  Update $\mathbf{z}_j \leftarrow \mathbf{z}_j - \eta \nabla_{\mathbf{z}_j} \mathcal{L}_{\text{diversity}}$ for $j = 1$ to M
end for

5.3 Experimental Evaluation

Datasets: We use the FLAME [Li et al., 2017a] registered head meshes from the CoMA [Ranjan et al., 2018] and D3DFACS [Cosker et al., 2011] datasets for training the Mesh-VAE, as well as for evaluating the proposed approach. Note that, other than the Mesh-VAE, our approach does not involve training any other modules. We split the two datasets into 80:10:10 train:val:test splits based on subject ID. We train the Mesh-VAE model using the combined training splits from the two datasets. During training, we augment the meshes with occlusion masks of random (contiguous) shapes at random locations. To evaluate our approach, we use the test split of the CoMA dataset [Ranjan et al., 2018] consisting of subjects that were excluded from training. Furthermore, we conduct a qualitative evaluation on the un-annotated images from the CelebA dataset [Liu et al., 2015]. For both datasets, the test images are artificially augmented with occlusions such as masks, glasses, and other random objects.

Implementation: We implement the Mesh-VAE as a fully convolutional graph neural network (GNN) based upon the MeshConv architecture presented in [Zhou et al., 2020b]. MeshConv [Zhou et al., 2020b] uses spatially varying convolution kernels to account for the irregularity of local mesh structures and was shown to outperform fixed kernel-based GNN approaches [Kipf and Welling, 2016, Defferrard et al., 2016, Morris et al., 2019, Veličković et al., 2017, Ranjan et al., 2018, Bouritsas et al., 2019] on reconstruction tasks. To train Mesh-VAE as a shape completion model, we augment the training meshes with random continuous masks covering 25-40% of the vertices. However, in practice, directly training the Mesh-VAE for inpainting is very challenging, especially with large degrees of occlusions. We adopt a curriculum learning [Bengio et al., 2009] approach to overcome this challenge and progressively introduce larger occlusions during the training process, i.e., we start with easier shape completion tasks and progressively increase its difficulty. We use a combination of ℓ_1-reconstruction, ℓ_1-Laplacian, and the KL-divergence losses to train the network. Note that we do not use partial shape completions fitted to occluded face images using either the FLAME [Li et al., 2017a] or our global+local model to train the Mesh-VAE, and instead use ground truth meshes to avoid any bias towards either shape model.

Occlusion	DECA [Feng et al., 2021]	FLAME [Li et al., 2017a]	Global+Local (Ours)
Glasses	57.83	47.89	**39.98**
Face-mask	61.18	30.37	**30.11**
Random	70.34	47.56	**38.27**
Overall	62.91	41.24	**35.85**

Table 5.1 Comparison of 3D reconstruction accuracy evaluated in terms of mean shape error (MSE) $\times 10^{-3}$.

Baselines: To evaluate the efficacy of Diverse3DFace in terms of diversity and robustness to occlusions, we compare against baselines such as FLAME [Li et al., 2017a], DECA [Feng et al., 2021], CFR-GAN [Ju et al., 2022], Occ3DMM [Egger et al., 2018] and Extreme3D [Tuán Trán et al., 2018] using publicly available implementations or pretrained models (wherever applicable). Due to the difficulty and unreliability in obtaining dense correspondence between FLAME and other mesh topologies, we perform a quantitative comparison only against methods based on the FLAME [Li et al., 2017a] topology. In other cases, we report qualitative comparisons based on face images with various occlusions patterns.

Metrics: The goal of this paper is to generate diverse yet realistic 3D reconstructions of occluded face images. Such an approach should have three desired qualities: 1) the reconstructed shapes should fit as accurately as possible to the visible regions, 2) the occluded regions should be diverse from each other, and 3) at least one of the reconstructed shapes should be very similar to the ground truth shape. There is no prior work on diverse 3D reconstruction, and as such, there are no established metrics. So we define the following three metrics to evaluate the aforementioned qualities: (1) **Closest Sample Error (CSE)**: the per-vertex ℓ_2-error between the ground-truth shape and the closest reconstructed shape (lower is better), (2) **Average Self Distance-Visible (ASD-V)**: the per-vertex ℓ_2-distance on the visible regions between a 3D completion and its closest neighbor, averaged across all the samples (lower is better), and (3) **Average Self Distance-Occluded (ASD-O)**: ASD on occluded regions (higher is better). These metrics are inspired by those defined for diverse trajectory forecasting [Yuan and Kitani, 2019].

5.3.1 Quantitative Results

5.3.1.1 Fitting on the Visible Regions

Tab. 5.1 reports the 3D reconstruction accuracy in terms of mean shape error (MSE) on artificially occluded test images from the CoMA dataset [Ranjan et al., 2018] for different approaches using the FLAME [Li et al., 2017a] topology. Across all occlusion types, our proposed global+local model reports the lowest MSE values. The large gap between FLAME (fitting) [Li et al., 2017a], DECA [Feng et al., 2021] and our approach demonstrates the necessity of region-specific model fitting for occlusion robustness.

5.3.2 Error Histogram Analysis

In Fig. 5.2, we plot the histograms of shape fitting errors (in terms of MSE) when the FLAME [Li et al., 2017a] and our global+local model are used to fit to partially occluded face images. One can observe that, while FLAME registers smaller errors (less than 10 MSE) on more number of samples than the global+local model, there are significantly more number of samples ($\sim 15\%$) where FLAME registers very high MSE errors (> 50 MSE) than the global+local model. One can conclude that our global+local model is more robust than the global FLAME model [Li et al., 2017a] on samples with challenging occlusions.

5.3.2.1 Diversity on the Occluded Regions

Due to the lack of existing diverse 3D reconstruction approaches, we formulate four baselines to evaluate the diversity performance of Diverse3DFace: 1) fitting FLAME on the visible parts plus DPP loss on the occluded parts (FLAME+DPP), 2) replace FLAME in (1) with our global+local model (Global+Local+DPP), 3) fitting global+local model followed by shape completions by the Mesh-VAE as per the learned distribution $p(\boldsymbol{S}_c, \boldsymbol{z}|\boldsymbol{S}_m)$ (Global+Local+VAE), and 4) replacing the global+local model with FLAME[Li et al., 2017a] in Diverse3DFace (FLAME+VAE+DPP). We report the quantitative metrics in Tab. 5.2. Across all occlusion types, FLAME+DPP and Global+Local+DPP report much higher *CSE* and *ASD-V,* and lower *ASD-O* than Diverse3DFace. Though Global+Local+VAE obtains lower *CSE* than Diverse3DFace, it does so at the cost of

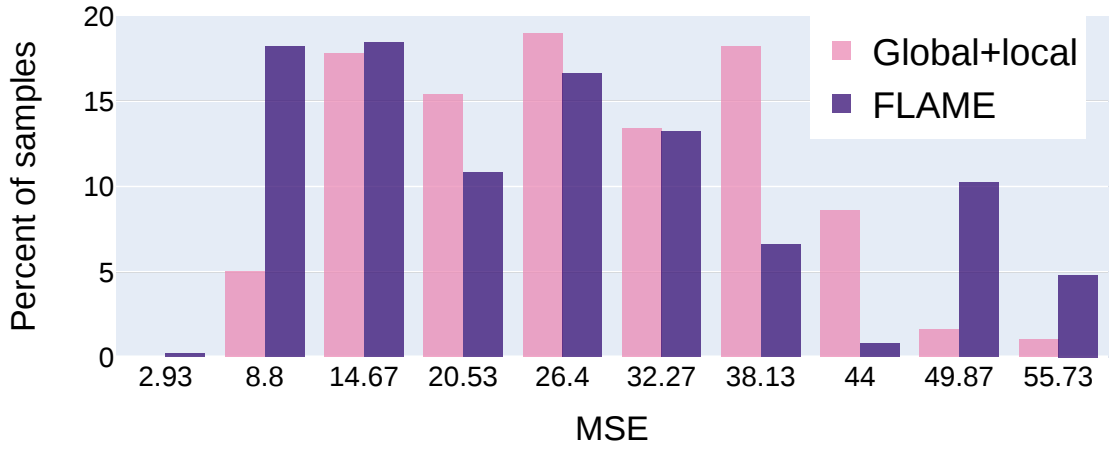


Figure 5.2 Histogram of MSE for shape fitting on occluded face images by FLAME [Li et al., 2017a] and our Global+local model.

reduced diversity in terms of *ASD-O*. FLAME+VAE+DPP reports better diversity metrics but at the cost of higher *CSE* errors. On the other hand, Diverse3DFace reports the lowest *ASD-V*, the highest *ASD-O*, and the second-lowest *CSE*, satisfying the three desired qualities mentioned earlier.

Since the CelebA dataset [Liu et al., 2015] is not labeled with groundtruth 3D shape, we do not compute the Closest Sample Distance (*CES*) on this dataset. As reported in **??**, our approach obtains the maximum *ASD-O* across all occlusion types, the lowest *ASD-V* for *Glasses*, as well as the second lowest (compared to Mesh-VAE) *ASD-V* for *Facemasks* and *Random* occlusions. This is further corroborated by the significantly higher *ASD-O/ASD-V* ratios reported by Diverse3DFace compared to the baselines. Compared to this, single-stage diversity fitting baselines *viz.* FLAME+DPP and Global+Local+DPP generate the lowest *ASD-O/ASD-V* ratios, signifying that the 3D reconstructions generated by these approaches are neither diverse on the occluded regions, nor consistent with respect to the visible regions. On the other hand, one-pass samples generated by Global+Local+VAE are consistent with the visible face as reported by low *ASD-V*, but not diverse on the occluded regions (low *ASD-O*).

These observations confirm our hypothesis that explicitly accounting for occlusions and op-

Occlusion Type	FLAME+DPP			Global+Local+DPP			Global+Local+VAE			FLAME+VAE+DPP			Global+Local+VAE+DPP (Ours)		
	CSE (↓)	**ASD-V** (↓)	**ASD-O** (↓)	**CSE** (↓)	**ASD-V** (↓)	**ASD-O** (↑)	**CSE** (↓)	**ASD-V** (↓)	**ASD-O** (↑)	**CSE** (↓)	**ASD-V** (↓)	**ASD-O** (↑)	**CSE** (↓)	**ASD-V** (↓)	**ASD-O** (↑)
Glasses	41.26	3.83	3.26	38.17	2.25	3.11	**32.88**	1.01	1.38	42.58	0.63	4.43	36.30	**0.61**	**4.50**
Face-mask	28.14	3.07	4.58	28.06	2.30	3.57	**25.95**	0.89	1.79	27.97	**0.61**	7.88	27.58	0.85	**7.89**
Random	43.12	3.61	4.06	38.85	2.59	3.51	**36.58**	0.97	1.61	43.00	0.78	5.44	39.11	**0.72**	**5.62**
Overall	36.81	3.61	4.06	34.55	2.35	3.39	**31.18**	0.95	1.59	37.45	0.77	5.92	33.71	**0.73**	**6.05**

Table 5.2 Evaluation of diverse reconstructions by the baselines *vs.* Diverse3DFace in terms of CSE, ASD-V and ASD-O (in order of 10^{-3}).

Occlusion Type	FLAME+DPP			Global+Local+DPP			Gloal+Local+VAE			Diverse3DFace (Ours)		
	ASD-V (↓)	**ASD-O** (↑)	$\frac{\textbf{ASD-O}}{\textbf{ASD-V}}$(↑)	**ASD-V** (↓)	**ASD-O** (↑)	$\frac{\textbf{ASD-O}}{\textbf{ASD-V}}$(↑)	**ASD-V** (↓)	**ASD-O** (↑)	$\frac{\textbf{ASD-O}}{\textbf{ASD-V}}$(↑)	**ASD-V** (↓)	**ASD-O** (↑)	$\frac{ASD-O}{ASD-V}$(↑)
Glasses	3.44	2.98	0.866	2.15	2.99	1.391	0.81	1.17	1.444	**0.68**	**3.56**	**5.235**
Face-mask	3.45	4.93	1.429	2.85	3.99	1.400	**0.75**	1.62	2.160	1.03	**7.47**	**7.252**
Random	4.12	4.23	1.027	3.17	3.84	1.211	**0.79**	1.29	1.633	0.83	**4.30**	**5.181**
Overall	3.86	4.44	1.150	3.03	3.88	1.281	**0.78**	1.41	1.808	0.90	**5.41**	**6.011**

Table 5.3 Quantitative evaluation of the diversity in 3D reconstruction of occluded faces from the CelebA dataset [Liu et al., 2015] between the baselines *vs.* Diverse3DFace in terms of the ASD-V and ASD-O metrics (in order of 10^{-3}) and the ratio between them.

timizing for diversity can lead to 3D reconstructions that are both more accurate (on the visible regions) and more geometrically diverse (on the occluded regions). Among the different occlusion types, we report the highest *ASD-O* for face-masks. These results are consistent with the fact that human faces have higher variability in the mouth and nose regions, which our approach is able to learn and reproduce.

5.3.3 Qualitative Results

5.3.3.1 Fitting on the Visible Regions

FLAME vs Global+Local PCA Model: In addition to the quantitative comparison done in Tab. 5.1, we qualitatively compare the occlusion robustness of the global FLAME [Li et al., 2017a] model *vs.* our global+local model. In Fig. 5.3, we show some failure cases of the FLAME [Li et al., 2017a] based fitting on severely occluded images. Notice the severe deformations on the FLAME [Li et al., 2017a] fitted outputs, especially around the mouth. In contrast, the fittings by our global+local models look more faithful and detailed with respect to the visible parts. These observations further support our claim that a global+local model-based fitting performs better than a global-model based fitting on occluded face images.

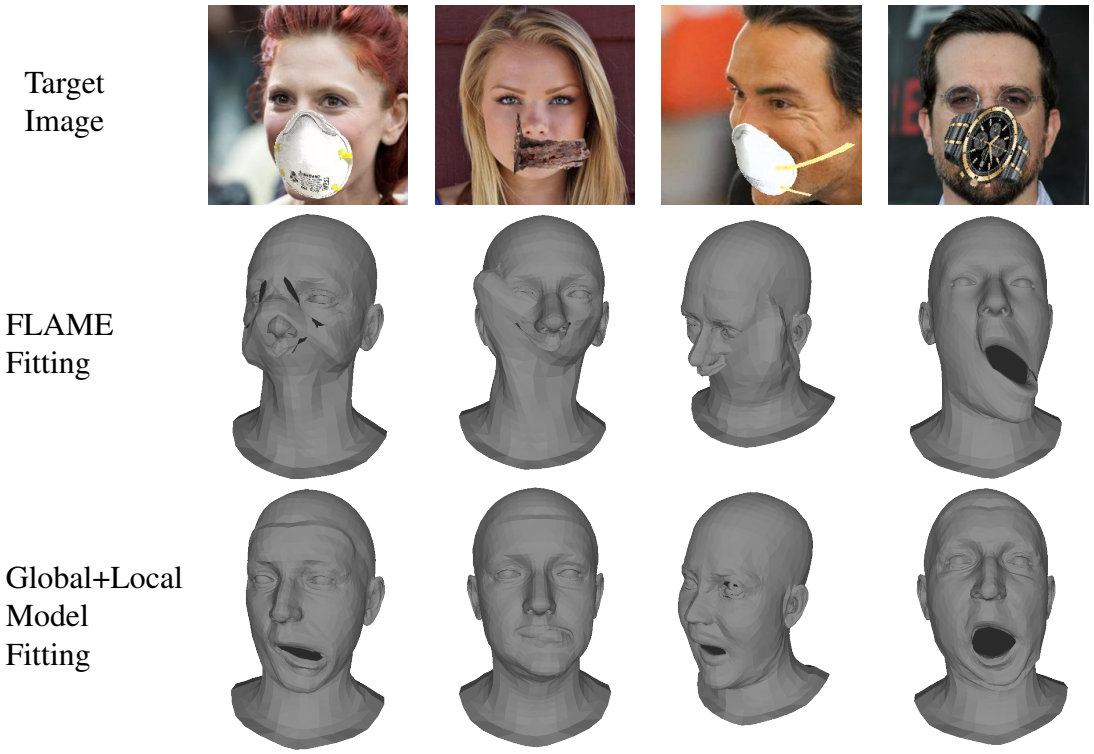


Figure 5.3 FLAME [Li et al., 2017a] based fitting (middle row) *vs.* our Global+Local fitting (last row) on occluded face images (top row).

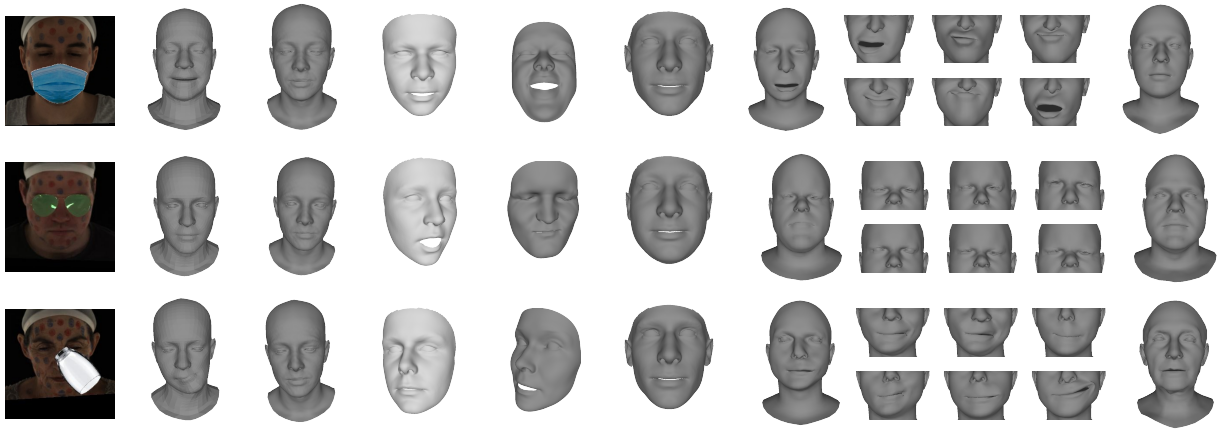


Figure 5.4 Qualitative evaluation on the CoMA dataset [Ranjan et al., 2018]: Reconstructed singular 3D meshes from the target image by the baselines *vs.* the diverse reconstructions (one full shape followed by six partial zoomed-in variations) from Diverse3DFace.

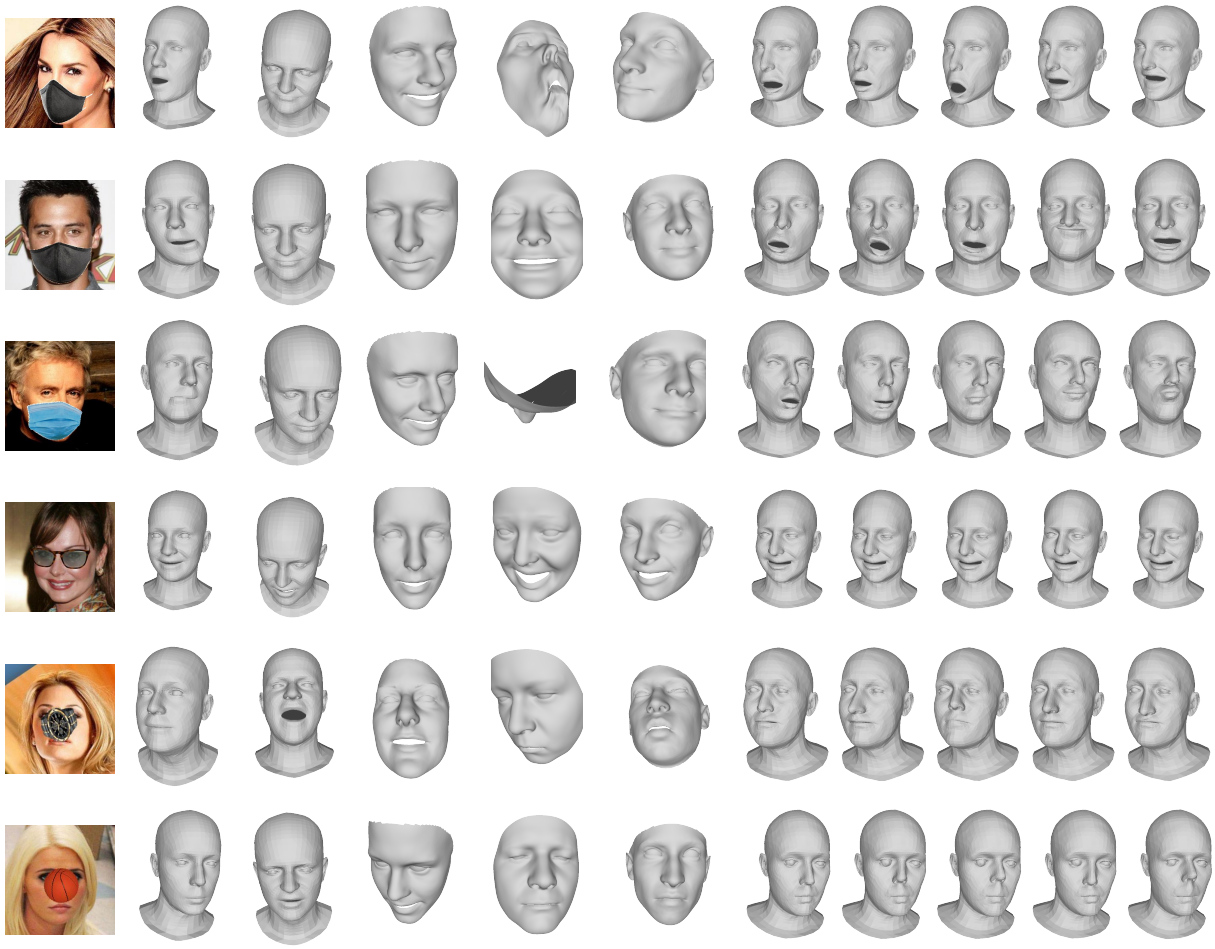


Figure 5.5 Qualitative evaluation on the CelebA dataset [Liu et al., 2015]: Reconstructed singular 3D meshes from the target image by the baselines *vs.* the diverse reconstructions from Diverse3DFace.

5.3.3.2 Diverse 3D Reconstructions

Fig. 5.4 shows qualitative results of 3D reconstruction on the artificially occluded CoMA [Ranjan et al., 2018] images. All the baselines can only generate a single 3D reconstruction w.r.t the target image. We observe that the reconstructions generated by Diverse3DFace look diverse yet plausible and visually more faithful to the ground truth in the visible regions. In comparison, FLAME-based fitting [Li et al., 2017a], and DECA [Feng et al., 2021] do not explicitly handle occlusions and generate soft and erroneous shapes. CFR-GAN [Ju et al., 2022] and Occ-3DMM [Egger et al., 2018] get the pose wrong in multiple instances. Extreme3D [Tuán Trán et al., 2018]

Figure 5.6 Set of 3D reconstructions by Diverse3DFace on real-world occluded face images.

generates visually better reconstructions of the visible parts of the face but gets the expression wrong in the second row. In Fig. 5.5, we show further visual comparisons on the occlusion-augmented images from the CelebA [Liu et al., 2015] dataset. Note that we do not have ground truth scans for these images. However, visual results suggest that the baselines, by being holistic models, do not explicitly exclude features from the occluded regions and often get incorrect poses and expressions on these images. Meanwhile, the reconstructions from Diverse3DFace look diverse on the occluded regions yet consistent w.r.t to the visible parts of the face.

5.3.4 Real-world Occlusions

We present examples of diverse 3D reconstructions by our approach on real-world occluded face images in Fig. 5.6. For these images, we inferred the occlusion mask using the face segmentation model by Nirkin *et al.* [Nirkin et al., 2018]. These results further demonstrate the efficacy of Diverse3DFace to generate diverse, yet plausible 3D reconstructions on real world occlusions ranging from glasses, scarf, facemasks, *etc*.

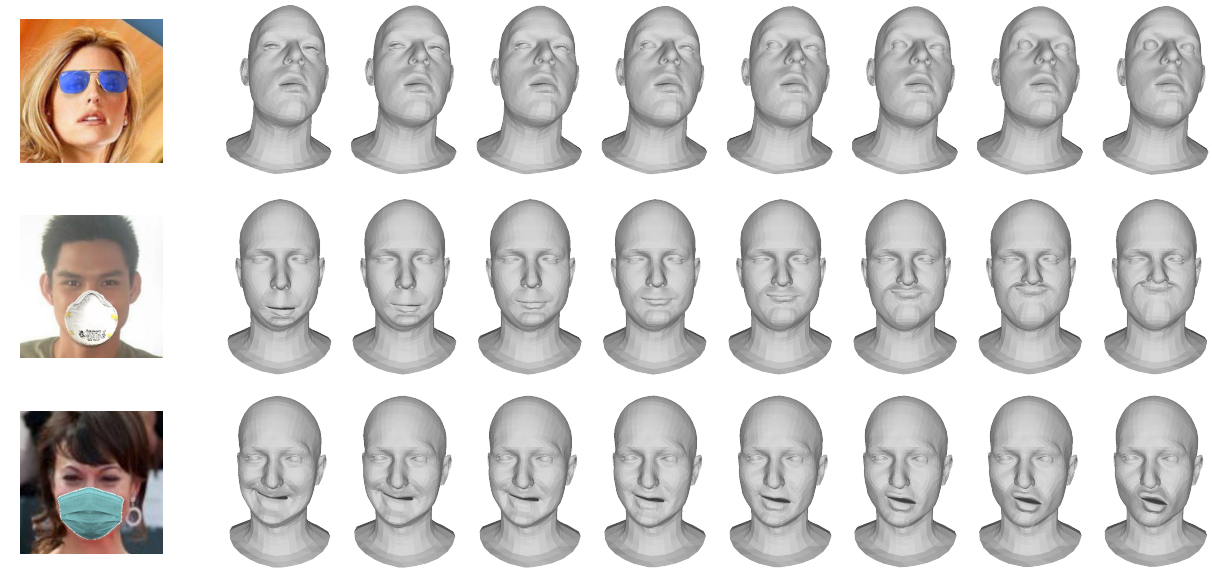


Figure 5.7 Controlled generated of diverse 3D reconstructions between two distinct modes. Diverse3DFace can be used to generate controlled diversity on the occluded regions by performing interpolation between two distinct shapes in the latent space.

5.3.4.1 Diversity Interpolations

A potential application of Diverse3DFace is to perform controlled diversification around an occluded region during 3D reconstruction. To do this, we can first generate a set of diverse 3D reconstructions for an occluded target image and then allow the user to select two distinct samples to perform interpolation in-between. We perform interpolation in the latent space: $\mathbf{z}(\alpha) = \alpha\mathbf{z}_1 + (1-\alpha)\mathbf{z}_2$. This affords the user control over the extent and type of diversity. We present examples of such interpolations in Fig. 5.7.

5.3.4.2 Moving the Occlusion Around the Face

In this section, we evaluate the diversity and robustness performance of Diverse3DFace to occlusions at different locations on the face. Fig. 5.8 shows the set of 3D reconstruction by Diverse3DFace when the occlusion moves around the face occupying the left cheek, mouth, the right cheek, center and the periocular (eye) regions of the face. Our method generates diverse, yet plausible set of 3D reconstructions for all the cases. We particularly note the high degree of diversity in expression that occurs when the mouth region is occluded, as is expected.

Figure 5.8 Qualitative evaluation of the diversity and robustness performance of Diverse3DFace to occlusions at different facial locations.

k \ n_σ	1	2	3	4	5
0.1	0.53	0.81	0.93	1.40	1.88
0.25	0.69	0.95	1.18	1.61	1.98
0.5	0.86	1.02	1.30	1.94	2.14
1	0.81	1.05	1.23	1.92	2.03
2	0.79	0.98	1.06	1.57	1.98

(a) ASD-V (↓)

k \ n_σ	1	2	3	4	5
0.1	3.63	4.92	5.62	7.17	8.64
0.25	4.13	6.37	7.65	8.18	10.73
0.5	5.98	8.25	9.16	11.19	14.53
1	5.18	7.89	8.84	10.72	12.96
2	4.42	6.68	7.40	9.78	12.21

(b) ASD-O (↑)

Table 5.4 Effect of the hyperparameters k and n_σ on the diversity metrics *ASD-V* and *ASD-O* on the CoMA dataset [Ranjan et al., 2018].

5.3.5 Ablation Study on Diversity Hyperparameters

The diversity generated by our approach is determined by the DPP loss:

$$L_{dpp} = -tr\left(\mathbf{I} - (\mathbf{L} + \mathbf{I})^{-1}\right). \tag{5.12}$$

Here, the DPP kernel entry for the i,j-th element is given by $L_{i,j} = q_i S_{i,j} q_j$, where q_i denotes the quality of element i, and $S_{i,j}$ represents the similarity between i and j. The DPP optimization tries to maximize the quality of each sample, while minimizing the similarity between distinct samples. As stated in the main paper, we control the similarity term $S_{i,j} = \exp\left(-\frac{k}{\text{median}_{i,j}(dist_{i,j})} dist_{i,j}\right)$ and the quality term $q_i = \exp(-\max(0, \mathbf{z}_i^T \mathbf{z}_i - n_\sigma\sqrt{d}))$ using two parameters k and n_σ, respectively. In Tab. 5.4, we study the effects of the two hyper-parameters k and n_σ on diversity as measured by the diversity metrics *ASD-V* and *ASD-O*. As shown in Tab. 5.4, we obtain maximum *ASD-V*, as well as, *ASD-O* at $k = 0.5$; whereas both metrics increase as n_σ increases. Thus, we set $k = 0.5$ in our experiments while we choose $n_\sigma = 3$ as a sweet spot between minimizing *ASD-V* and maximizing *ASD*-O. The user can change the value of n_σ to tweak the diversity-realism trade-off.

5.4 Conclusion

We proposed Diverse3DFace, an approach to reconstruct diverse yet plausible 3D reconstructions corresponding to a single occluded face image. Our approach was motivated by the fact that, in the presence of occlusions, a distribution of plausible 3D reconstructions is more desirable than a single unique solution. We proposed a three-step solution that first fits a robust partial shape using an ensemble of global+local PCA models, maps it to a latent space, and iteratively optimizes

the embeddings to promote diversity in the occluded parts while retaining fidelity with respect to the visible parts of the face. Experimental evaluation across multiple occlusion types and datasets show the efficacy of Diverse3DFace, both in terms of robustness and diversity, compared to multiple baselines. To our knowledge, this is the first approach that generates a distribution of diverse 3D reconstructions of a single occluded face image.

A limitation of the proposed approach is its dependence on the robustness of the global+local fitting in the first step for further diverse completions. Although such a locally disentangled fitting demonstrably performs better than a global model fitting, it may still be affected in cases where the initial landmark or face-mask estimates are wrong.

CHAPTER 6

FUTURE EXTENSIONS

So far, we have focussed on how 3D-aware generative modeling can improve face inpainting, and controlled face generation and editing. We also studied ways to make monocular 3D face reconstruction generate robust and diverse solutions in the presence of occlusions. These works have natural extensions that we now propose.

6.1 Generating Diverse Textured 3D Reconstructions from a Single Occluded Face Image

In Chapter 5, we generated a distribution of diverse, but realistic 3D reconstructions corresponding to an occluded face image such that we retain fidelity with respect to the visible parts, and allow for realistic diversity on the occluded parts of the face. This work was motivated by the observation that, in the presence of occlusion, no one 3D reconstruction can be said to be the correct one. However, one can naturally extend this reasoning to the domain of appearance too. That is, it is possible for the occluded part to vary in shape, expression, and albedo, while global factors like illumination can remain constant. Reconstruction models therefore need to account for diversity from several perspectives. The utility of such an algorithm will not just be restricted to occlusion robust 3D reconstruction, but will also extend to editing specific parts of face in 3D, both in shape as well as appearance. While one way of attempting this can be simply extending Diverse3DFace to include texture by estimating a partial texture with respect to the visible regions, followed by diverse completions using a texture-VAE, yet another way could be by leveraging the power of diffusion models [Ho et al., 2020]. Diffusion models have gained much traction in the recent years in the domain of generative modelling [Dhariwal and Nichol, 2021, Lugmayr et al., 2022], and inherently support diversity, which stems out of its stochastic sampling approach. This way, we can model a joint distribution of both shape and texture and sample diverse 3D reconstructions from this, conditioned on the partial estimates.

We show a proposed overview of such an approach in Fig. 6.1. It consists of three components: (i) a partial 3D reconstruction component, (ii) a generative prior component, and (iii) an explicit diversification component. The partial 3D reconstruction involves reconstructing just the visible

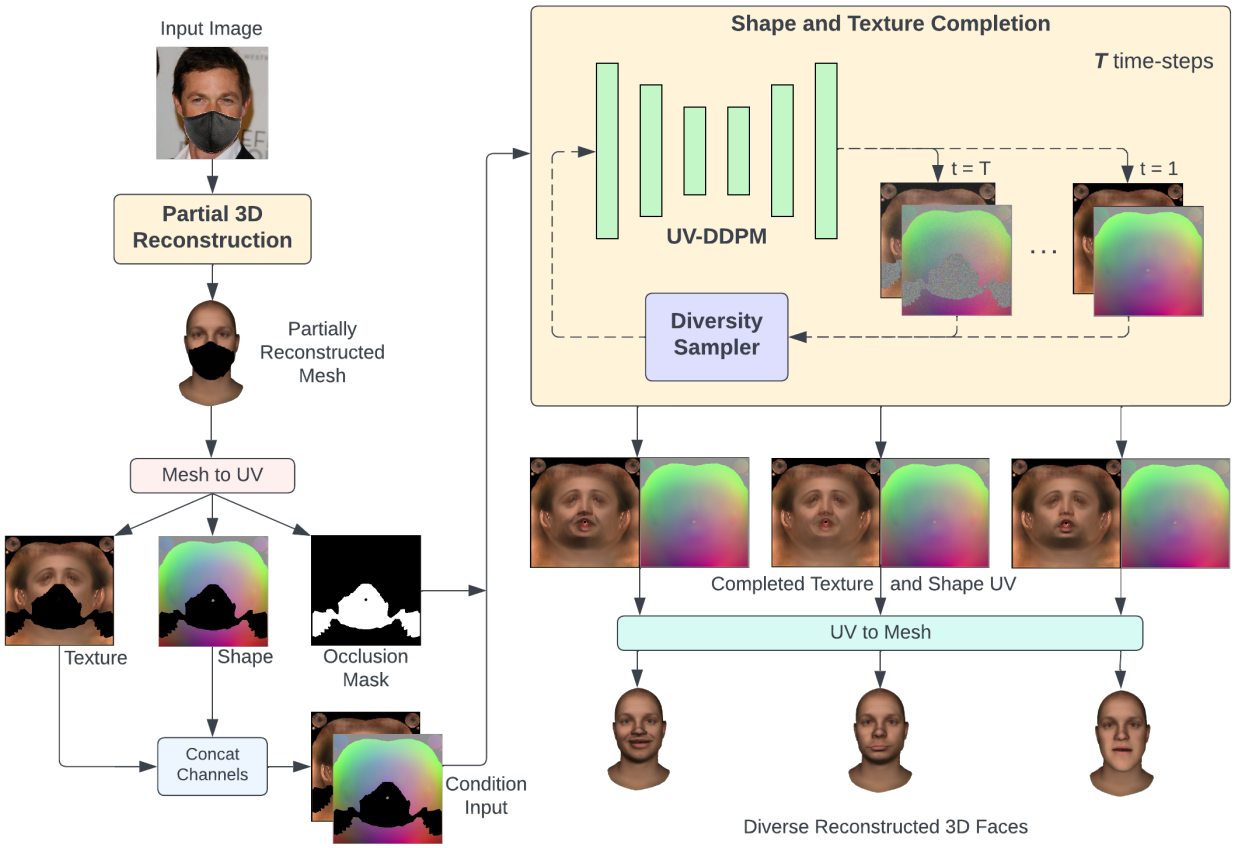


Figure 6.1 Overview of the proposed DDPM powererd Diverse3DFace (DivFusionFace). After obtaining an initial partial 3D reconstruction, we propose to transform our mesh to its UV representation, and perform diverse shape and texture completion using a UV-DDPM and diverse sampler. The completed UVs can then be transformed back to their mesh representation.

part, and can be done with using an occlusion-robust 3D reconstruction algorithm such as our global+local model (see Sec. 5.2.1). Then the partial shape and texture in their UV representations are inpainted using a DDPM. To promote diversity, we can further replace the standard sampler in DDPM with a diversity-aware reverse diffusion sampler. We achieve this by incorporating the idea of DPP kernel [Kulesza and Taskar, 2012] into the reverse process of a DDPM, such that generated samples take diverse sampling trajectories from each other. This will enable our proposed approach to generate samples with controllable levels of diversity, both in terms of texture and shape.

6.2 High-Resolution Diversity-Oriented 3DFaceFill

We have shown that 3DFaceFill [Dey and Boddeti, 2022a] can inpaint partial face images while maintaining the structural integrity of human face, owing to it incorporating explicit face 3D

priors. However, the implementation in Chapter 3 has two main limitations: (i) it is limited by the resolution of the underlying 3D model which in the case of 3DFaceFill is the Basel Face Model (BFM) [Paysan et al., 2009], (ii) it does not model the regions not included in the underlying 3D model, such as hair and teeth, and (iii) it generates a single inpainted solution, which does not satisfy the desired property of diversity in the presense of missing information, as recognized in this thesis. A future extension of this work should try to fix these limitations.

For the first and second limitation of limited resolution, we can adopt on the following two approaches:

1. We can add a second stage to the pipeline that takes in the low-resolution inpainted image from the first stage, and super-resolves it into a higher resolution image similar to the styled-generator in Chapter 4; or

2. We can employ a higher-resolution face 3D model that also includes inner mouth such as the UHM model [Ploumpis et al., 2020], instead of the BFM model [Paysan et al., 2009]

Diverse inpainting with respect to geometry can be achieved by plugging in our approach Diverse3DFace [Dey and Boddeti, 2022b], as against the 3DMM model used in 3DFaceFill, for diverse occlusion-aware 3D reconstruciton. Moreover, the extension proposed in Sec. 6.1 would enable diversity in both geometry and appearance.

6.3 Extensions to CoLa-SDF

While our presented approach CoLa-SDF can generate high fidelity 3D faces with high degree of control over shape, albedo, illumination, hairstyle and pose, it can be made even more practically useful with some extensions. We outline some of these below:

1. **Text-based Control:** Text-conditioned generative modeling have been becoming more popular with the introduction of CLIP [Radford et al., 2021]. Using CLIP, we can train neural networks to control the latent codes of CoLa-SDF corresponding to these attributes, enabling text controlled face generation in 3D. We can also extend this to Diverse3DFace. Whereas diversity is introduced in a stochastic way using DPP [Kulesza and Taskar, 2012]

in Diverse3DFace, a future extension can make shape completion on the occluded parts conditioned on textual inputs.

2. **Explicit Identity-Preservation:** A use-case for CoLa-SDF could be identity-preserving reconstruction and editing of faces in 3D. In the current implementation, CoLa-SDF preserves identity implicitly when editing illumination, pose, and hair/background regions. However, this is not enforced using an identity preserving loss. A future extension can be trained simply by adding an identity preserving loss, followed by a thorough face recognition evaluation, can demonstrate important practical applications such as in virtual avatars, virtual meetings and others.

3. **Semantic Hair-control:** Though CoLa-SDF can edit hair and background independent of the rest of the facial attributes, we cannot explicitly control semantic attributes of hair such as length, style, and color. This can be achieved 1) by using carefully sampled latent codes corresponding to specific attributes in the training set, or 2) by using a semantic 3D model of human hair such as [Wu et al., 2022].

CHAPTER 7

CONCLUSION

In this thesis, we explored the possibilities and opportunities that come with 3D modeling of faces to tasks such as face inpainting and controlled 3D face generation. We also studied the problem of occlusions in 3D reconstruction and clamied that robustness, diversity and maintaining structural integrity of the face should be the cornerstone criteria by which such occlusion-aware models should be evaluated.

Towards 3D-aware face inpainting, we proposed 3DFaceFill [Dey and Boddeti, 2022a], which was driven by the hypothesis that 3D disentanglement of face image into 3D shape, 3D pose, albedo and illumination, followed by albedo inpainting in the UV representation, as opposed to 2D pixel representation, will allow us to effectively leverage the power of 3D correspondence and ultimately lead to face completions that are geometrically and photometrically more accurate. Experimental evaluation across multiple datasets and against multiple baselines show that face completions from 3DFaceFill are significantly better, both qualitatively and quantitatively, under large variations in pose, illumination, shape and appearance, which validate our hypothesis.

To enable controllable generation of 3D faces, we proposed CoLa-SDF that combines the disentangled controllability of nonlinear 3DMM approaches with the high fidelity of implicit 3D-GANs. Building upon the architecture of StyleSDF [Or-El et al., 2022], we enforce the latent space to match the physical parameters of the nonlinear 3D morphable model MOST-GAN [Medin et al., 2022]. We also enforced disentangled control of hair and background, a feat we believe is a first of its kind. We demonstrate high-fidelity image synthesis and subsequent 3D manipulation with full control over the over the 3D shape, pose, albedo, illumination and hairstyle of the generated face.

To address the challenge of facial occlusions in single view 3D face reconstruction, we proposed Diverse3DFace [Dey and Boddeti, 2022b], which reconstructs diverse yet plausible 3D models corresponding to a single occluded face image. Our approach was motivated by the three fold criteria of occlusion robustness, diversity and maintaining structural integrity of faces. We presented a three-step solution of first fitting a robust partial shape using an ensemble of global+local

PCA models, mapping it to the latent space of a mesh-VAE, and iteratively optimizing the embeddings to promote diversity in the occluded parts, while retaining fidelity with respect to the visible parts of the face. Experimental evaluation across multiple occlusion types and datasets show the efficacy of Diverse3DFace compared to multiple baselines, both in terms of robustness and diversity.

Limitations: Despite the improvements our proposed approaches have over the traditional approaches in terms of face inpainting, controllable face generation, and occlusion-aware 3D reconstruction, our approaches have certain limitations. 3DFaceFill is based upon the template based BFM model [Paysan et al., 2009] which doesn't include inner mouth cavity and hair, and is limited in its resolution. These limitations, thus, transfer to 3DFaceFill too. In Chapter 6, we propose future enhancements to overcome these limitations including adopting a different 3D model such as UHM [Ploumpis et al., 2020], and adding a refiner or a subsequent super-resolver module. 3DFaceFill also generates a singular, and not diverse solutions. We can replace the underlying 3D reconstruction algorithm with the proposed Diverse3DFace to enable diverse completions.

Our approach for controllable 3D face generation, CoLa-SDF, sometimes produce artifacts when sampling from beyond three standard deviations from the mean MOST-GAN parameters. Further, when interpolating between two hairstyles, it sometimes generate incomplete hats. Both these effects may be due a lack of such examples in the FFHQ dataset [Karras et al., 2020] on which it is trained. Fine tuning on a more diverse face dataset, or weighted sampling to favor sampling extreme examples more often during training may address these challenges.

A limitation of Diverse3DFace is its dependence on the robustness of the global+local fitting in the first step for further diverse completions. Although such a locally disentangled fitting demonstrably performs better than a global model fitting, it may still be affected in cases where the initial landmark or face-mask estimates are wrong. Also, extending Diverse3DFace to include texture and model diversity in both shape and texture is a desirable objective. To address this, we have proposed future extensions in Sec. 6.1.

BIBLIOGRAPHY

[Athar et al., 2022] Athar, S., Xu, Z., Sunkavalli, K., Shechtman, E., and Shu, Z. (2022). Rignerf: Fully controllable neural 3d portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20364–20373.

[Bahat and Michaeli, 2020] Bahat, Y. and Michaeli, T. (2020). Explorable super resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2716–2725.

[Barnes et al., 2009] Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24.

[Bengio et al., 2009] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *International Conference on Machine Learning*.

[Bertalmio et al., 2000] Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424.

[Blanz and Vetter, 1999] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194.

[Bouritsas et al., 2019] Bouritsas, G., Bokhnyak, S., Ploumpis, S., Bronstein, M., and Zafeiriou, S. (2019). Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7213–7222.

[Chan et al., 2022] Chan, E. R., Lin, C. Z., Chan, M. A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L. J., Tremblay, J., Khamis, S., et al. (2022). Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133.

[Chan et al., 2021] Chan, E. R., Monteiro, M., Kellnhofer, P., Wu, J., and Wetzstein, G. (2021). pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809.

[Che et al., 2016] Che, T., Li, Y., Jacob, A. P., Bengio, Y., and Li, W. (2016). Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*.

[Chen et al., 2017a] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.

[Chen et al., 2017b] Chen, Y.-A., Chen, W.-C., Wei, C.-P., and Wang, Y.-C. F. (2017b). Occlusion-aware face inpainting via generative adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1202–1206. IEEE.

[Clevert et al., 2015] Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

[Cosker et al., 2011] Cosker, D., Krumhuber, E., and Hilton, A. (2011). A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *2011 international conference on computer vision*, pages 2296–2303. IEEE.

[Criminisi et al., 2004] Criminisi, A., Pérez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212.

[Defferrard et al., 2016] Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852.

[Deng et al., 2018] Deng, J., Cheng, S., Xue, N., Zhou, Y., and Zafeiriou, S. (2018). Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7093–7102.

[Deng et al., 2019] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699.

[Deng et al., 2020] Deng, Y., Yang, J., Chen, D., Wen, F., and Tong, X. (2020). Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5154–5163.

[Dey and Boddeti, 2022a] Dey, R. and Boddeti, V. N. (2022a). 3dfacefill: An analysis-by-synthesis approach to face completion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1586–1595.

[Dey and Boddeti, 2022b] Dey, R. and Boddeti, V. N. (2022b). Generating diverse 3d reconstructions from a single occluded face image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1547–1557.

[Dhariwal and Nichol, 2021] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794.

[Egger et al., 2018] Egger, B., Schönborn, S., Schneider, A., Kortylewski, A., Morel-Forster, A., Blumer, C., and Vetter, T. (2018). Occlusion-aware 3d morphable models and an illumination prior for face image analysis. *International Journal of Computer Vision*, 126(12):1269–1287.

[Egger et al., 2020] Egger, B., Smith, W. A., Tewari, A., Wuhrer, S., Zollhoefer, M., Beeler, T., Bernard, F., Bolkart, T., Kortylewski, A., Romdhani, S., et al. (2020). 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38.

[Elfeki et al., 2019] Elfeki, M., Couprie, C., Riviere, M., and Elhoseiny, M. (2019). Gdpp: Learning diverse generations using determinantal point processes. In *International Conference on Machine Learning*, pages 1774–1783. PMLR.

[Feng et al., 2021] Feng, Y., Feng, H., Black, M. J., and Bolkart, T. (2021). Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (TOG)*, 40(4):1–13.

[Gecer et al., 2019] Gecer, B., Ploumpis, S., Kotsia, I., and Zafeiriou, S. (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 (CVPR)*.

[Gerig et al., 2018] Gerig, T., Morel-Forster, A., Blumer, C., Egger, B., Luthi, M., Schönborn, S., and Vetter, T. (2018). Morphable face models-an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82. IEEE.

[Ghosh et al., 2018] Ghosh, A., Kulharia, V., Namboodiri, V. P., Torr, P. H., and Dokania, P. K. (2018). Multi-agent diverse generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[Grassal et al., 2022] Grassal, P.-W., Prinzler, M., Leistner, T., Rother, C., Nießner, M., and Thies, J. (2022). Neural head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18653–18664.

[Gropp et al., 2020] Gropp, A., Yariv, L., Haim, N., Atzmon, M., and Lipman, Y. (2020). Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*.

[Gross et al., 2010] Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. (2010). Multi-pie. *Image and Vision Computing*, 28(5):807–813.

[Gu et al., 2021] Gu, J., Liu, L., Wang, P., and Theobalt, C. (2021). Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*.

[Gulrajani et al., 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777.

[Hays and Efros, 2007] Hays, J. and Efros, A. A. (2007). Scene completion using millions of photographs. *ACM Transactions on Graphics (ToG)*, 26(3):4–es.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Heusel et al., 2017] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

[Hong et al., 2021] Hong, Y., Peng, B., Xiao, H., Liu, L., and Zhang, J. (2021). Headnerf: A real-time nerf-based parametric head model. *arXiv preprint arXiv:2112.05637*.

[Iizuka et al., 2017] Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14.

[Isola et al., 2017] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

[Ju et al., 2022] Ju, Y.-J., Lee, G.-H., Hong, J.-H., and Lee, S.-W. (2022). Complete face recovery gan: Unsupervised joint face rotation and de-occlusion from a single-view image. In *WACV*.

[Juefei-Xu et al., 2018] Juefei-Xu, F., Dey, R., Boddeti, V. N., and Savvides, M. (2018). Rankgan: a maximum margin ranking gan for generating faces. In *Asian Conference on Computer Vision*, pages 3–18. Springer.

[Karras et al., 2021] Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. (2021). Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863.

[Karras et al., 2019] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.

[Karras et al., 2020] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119.

[Kendall and Gal, 2017] Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584.

[Kim et al., 2018] Kim, H., Zollhöfer, M., Tewari, A., Thies, J., Richardt, C., and Theobalt, C. (2018). Inversefacenet: Deep monocular inverse face rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4625–4634.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[Kulesza and Taskar, 2012] Kulesza, A. and Taskar, B. (2012). Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*.

[Lee et al., 2020] Lee, C.-H., Liu, Z., Wu, L., and Luo, P. (2020). Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Li et al., 2023] Li, C., Morel-Forster, A., Vetter, T., Egger, B., and Kortylewski, A. (2023). Robust model-based face reconstruction through weakly-supervised outlier segmentation. In *36th IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE.

[Li et al., 2017a] Li, T., Bolkart, T., Black, M. J., Li, H., and Romero, J. (2017a). Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17.

[Li et al., 2020a] Li, X., Hu, G., Zhu, J., Zuo, W., Wang, M., and Zhang, L. (2020a). Learning symmetry consistent deep cnns for face completion. *IEEE Transactions on Image Processing*, 29:7641–7655.

[Li et al., 2017b] Li, Y., Liu, S., Yang, J., and Yang, M.-H. (2017b). Generative face completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3911–3919.

[Li et al., 2020b] Li, Z., Hu, Y., He, R., and Sun, Z. (2020b). Learning disentangling and fusing networks for face completion under structured occlusions. *Pattern Recognition*, 99:107073.

[Lin et al., 2017] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

[Liu et al., 2018] Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100.

[Liu et al., 2015] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[Lugmayr et al., 2022] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471.

[Maas et al., 2013] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

[Macchi, 1975] Macchi, O. (1975). The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122.

[Medin et al., 2022] Medin, S. C., Egger, B., Cherian, A., Wang, Y., Tenenbaum, J. B., Liu, X., and Marks, T. K. (2022). Most-gan: 3d morphable stylegan for disentangled face image manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1962–1971.

[Mescheder et al., 2018] Mescheder, L., Geiger, A., and Nowozin, S. (2018). Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR.

[Mildenhall et al., 2020] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer.

[Miyato et al., 2018] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.

[Morris et al., 2019] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. (2019). Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609.

[Neumann et al., 2013] Neumann, T., Varanasi, K., Wenger, S., Wacker, M., Magnor, M., and Theobalt, C. (2013). Sparse localized deformation components. *ACM Transactions on Graphics (TOG)*, 32(6):1–10.

[Niemeyer and Geiger, 2021] Niemeyer, M. and Geiger, A. (2021). Giraffe: Representing scenes as compositional generative neural feature fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Nirkin et al., 2018] Nirkin, Y., Masi, I., Tuan, A. T., Hassner, T., and Medioni, G. (2018). On face segmentation, face swapping, and face perception. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 98–105. IEEE.

[Or-El et al., 2022] Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, J. J., and Kemelmacher-Shlizerman, I. (2022). Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513.

[Parkhi et al., 2015] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *British Machine Vision Conference*.

[Pathak et al., 2016] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544.

[Paysan et al., 2009] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. (2009). A 3d face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301. Ieee.

[Ploumpis et al., 2020] Ploumpis, S., Ververas, E., O'Sullivan, E., Moschoglou, S., Wang, H., Pears, N., Smith, W., Gecer, B., and Zafeiriou, S. P. (2020). Towards a complete 3d morphable model of the human head. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Radford et al., 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

[Ramamoorthi and Hanrahan, 2001] Ramamoorthi, R. and Hanrahan, P. (2001). An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500.

[Ramon et al., 2021] Ramon, E., Triginer, G., Escur, J., Pumarola, A., Garcia, J., Giro-i Nieto, X., and Moreno-Noguer, F. (2021). H3d-net: Few-shot high-fidelity 3d head reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5620–5629.

[Ranjan et al., 2018] Ranjan, A., Bolkart, T., Sanyal, S., and Black, M. J. (2018). Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pages 725–741.

[Ravi et al., 2020] Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., and Gkioxari, G. (2020). Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[Saito et al., 2016] Saito, S., Li, T., and Li, H. (2016). Real-time facial segmentation and performance capture from rgb input. In *European conference on computer vision*, pages 244–261. Springer.

[Sanyal et al., 2019] Sanyal, S., Bolkart, T., Feng, H., and Black, M. J. (2019). Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7763–7772.

[Schwartz et al., 2020] Schwartz, G., Wei, S.-E., Wang, T.-L., Lombardi, S., Simon, T., Saragih, J., and Sheikh, Y. (2020). The eyes have it: An integrated eye and face model for photorealistic facial animation. *ACM Trans. Graph.*, 39(4).

[Schwarz et al., 2020] Schwarz, K., Liao, Y., Niemeyer, M., and Geiger, A. (2020). Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166.

[Sengupta et al., 2018] Sengupta, S., Kanazawa, A., Castillo, C. D., and Jacobs, D. W. (2018). Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6296–6305.

[Shocher et al., 2019] Shocher, A., Bagon, S., Isola, P., and Irani, M. (2019). Ingan: Capturing and retargeting the" dna" of a natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4492–4501.

[Shu et al., 2017] Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., and Samaras, D. (2017). Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5541–5550.

[Sitzmann et al., 2020] Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473.

[Sohn et al., 2015] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491.

[Song et al., 2019a] Song, L., Cao, J., Song, L., Hu, Y., and He, R. (2019a). Geometry-aware face completion and editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2506–2513.

[Song et al., 2019b] Song, L., Gong, D., Li, Z., Liu, C., and Liu, W. (2019b). Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 773–782.

[Srivastava et al., 2017] Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. (2017). Veegan: Reducing mode collapse in gans using implicit variational learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3310–3320.

[Sun et al., 2022] Sun, J., Wang, X., Zhang, Y., Li, X., Zhang, Q., Liu, Y., and Wang, J. (2022). Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682.

[Suzuki et al., 2016] Suzuki, M., Nakayama, K., and Matsuo, Y. (2016). Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891*.

[Tewari et al., 2020a] Tewari, A., Elgharib, M., Bernard, F., Seidel, H.-P., Pérez, P., Zollhöfer, M., and Theobalt, C. (2020a). Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)*, 39(6):1–14.

[Tewari et al., 2020b] Tewari, A., Elgharib, M., Bharaj, G., Bernard, F., Seidel, H.-P., Pérez, P., Zollhofer, M., and Theobalt, C. (2020b). Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151.

[Tewari et al., 2022] Tewari, A., Pan, X., Fried, O., Agrawala, M., Theobalt, C., et al. (2022). Disentangled3d: Learning a 3d generative model with disentangled geometry and appearance from monocular images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1516–1525.

[Tewari et al., 2017] Tewari, A., Zollhofer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., and Theobalt, C. (2017). Mofa: Model-based deep convolutional face autoencoder for unsupervised

monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283.

[Tran et al., 2019] Tran, L., Liu, F., and Liu, X. (2019). Towards high-fidelity nonlinear 3d face morphable model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1126–1135.

[Tran and Liu, 2018] Tran, L. and Liu, X. (2018). Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355.

[Tran and Liu, 2019] Tran, L. and Liu, X. (2019). On learning 3d face morphable model from in-the-wild images. *IEEE transactions on pattern analysis and machine intelligence*.

[Tripathy et al., 2021] Tripathy, S., Kannala, J., and Rahtu, E. (2021). Facegan: Facial attribute controllable reenactment gan. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1329–1338.

[Tuan Tran et al., 2017] Tuan Tran, A., Hassner, T., Masi, I., and Medioni, G. (2017). Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5163–5172.

[Tuán Trán et al., 2018] Tuán Trán, A., Hassner, T., Masi, I., Paz, E., Nirkin, Y., and Medioni, G. (2018). Extreme 3d face reconstruction: Seeing through occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3935–3944.

[Veličković et al., 2017] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

[Wang et al., 2022] Wang, C., Chai, M., He, M., Chen, D., and Liao, J. (2022). Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844.

[Wang et al., 2020] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. (2020). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*.

[Wang et al., 2021] Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., and Wang, W. (2021). Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*.

[Wang et al., 2018] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807.

[Wei et al., 2019] Wei, H., Liang, S., and Wei, Y. (2019). 3d dense face alignment via graph convolution networks. *arXiv preprint arXiv:1904.05562*.

[Wu et al., 2022] Wu, K., Ye, Y., Yang, L., Fu, H., Zhou, K., and Zheng, Y. (2022). Neuralhdhair: Automatic high-fidelity hair modeling from a single image using implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535.

[Wu et al., 2020] Wu, S., Rupprecht, C., and Vedaldi, A. (2020). Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10.

[Wu and He, 2018] Wu, Y. and He, K. (2018). Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.

[Xia et al., 2022] Xia, W., Zhang, Y., Yang, Y., Xue, J.-H., Zhou, B., and Yang, M.-H. (2022). Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Yang et al., 2019] Yang, D., Hong, S., Jang, Y., Zhao, T., and Lee, H. (2019). Diversity-sensitive conditional generative adversarial networks. *arXiv preprint arXiv:1901.09024*.

[Yang et al., 2022] Yang, G., Vo, M., Neverova, N., Ramanan, D., Vedaldi, A., and Joo, H. (2022). Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2863–2873.

[Yeh et al., 2017] Yeh, R. A., Chen, C., Yian Lim, T., Schwing, A. G., Hasegawa-Johnson, M., and Do, M. N. (2017). Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493.

[Yenamandra et al., 2021] Yenamandra, T., Tewari, A., Bernard, F., Seidel, H.-P., Elgharib, M., Cremers, D., and Theobalt, C. (2021). i3dmm: Deep implicit 3d morphable model of human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12803–12813.

[Yu et al., 2018] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2018). Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514.

[Yu et al., 2019] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2019). Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480.

[Yuan and Park, 2019] Yuan, X. and Park, I. K. (2019). Face de-occlusion using 3d morphable model and generative adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10062–10071.

[Yuan and Kitani, 2019] Yuan, Y. and Kitani, K. (2019). Diverse trajectory forecasting with determinantal point processes. *arXiv preprint arXiv:1907.04967*.

[Yuan and Kitani, 2020] Yuan, Y. and Kitani, K. (2020). Dlow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision*, pages 346–364. Springer.

[Yuan et al., 2022] Yuan, Y.-J., Sun, Y.-T., Lai, Y.-K., Ma, Y., Jia, R., and Gao, L. (2022). Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364.

[Zhang et al., 2018a] Zhang, J., Zhan, R., Sun, D., and Pan, G. (2018a). Symmetry-aware face completion with generative adversarial networks. In *Asian Conference on Computer Vision*, pages 289–304. Springer.

[Zhang and Samaras, 2006] Zhang, L. and Samaras, D. (2006). Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):351–363.

[Zhang et al., 2018b] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018b). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

[Zhang et al., 2017] Zhang, S., He, R., Sun, Z., and Tan, T. (2017). Demeshnet: Blind face inpainting for deep meshface verification. *IEEE Transactions on Information Forensics and Security*, 13(3):637–647.

[Zheng et al., 2019a] Zheng, C., Cham, T.-J., and Cai, J. (2019a). Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447.

[Zheng et al., 2019b] Zheng, C., Cham, T.-J., and Cai, J. (2019b). Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1438–1447.

[Zhou et al., 2020a] Zhou, T., Ding, C., Lin, S., Wang, X., and Tao, D. (2020a). Learning oracle attention for high-fidelity face completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7680–7689.

[Zhou et al., 2020b] Zhou, Y., Wu, C., Li, Z., Cao, C., Ye, Y., Saragih, J., Li, H., and Sheikh, Y. (2020b). Fully convolutional mesh autoencoder using efficient spatially varying kernels. *arXiv preprint arXiv:2006.04325*.

[Zhu et al., 2017] Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., and Shechtman, E. (2017). Multimodal image-to-image translation by enforcing bi-cycle consistency. In *Advances in neural information processing systems*, pages 465–476.

[Zhu et al., 2016] Zhu, X., Lei, Z., Liu, X., Shi, H., and Li, S. Z. (2016). Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155.

[Zollhöfer et al., 2018] Zollhöfer, M., Thies, J., Garrido, P., Bradley, D., Beeler, T., Pérez, P., Stamminger, M., Nießner, M., and Theobalt, C. (2018). State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer Graphics Forum*, volume 37, pages 523–550. Wiley Online Library.

APPENDIX A 3DFACEFILL

A.1 Generalization Performance of 3DFaceFill on In-the-Wild Images downloaded from the Internet

To compare the generalization performance of different methods, we evaluate face completion on a small dataset of ~ 50 in-the-wild face images downloaded from the internet[1] (referred to as Internet). We report the quantitative metrics in Table 3.1, where one can see significant margins between 3DFaceFill and the closest baselines across all the three metrics, demonstrating the better generalization performance of our proposed method. Fig. A.1 shows qualitative comparison on a small sample where 3DFaceFill generates more realistic completions, thanks to the explicit imposition of 3D face priors. This shows that the principles behind 3DFaceFill can improve the generalization performance of image completion approaches on structured objects such as faces.

A.2 Further Qualitative Results on Pose and Illumination Varying Images

We present further face completion on the pose and illumination varying images from the MultiPIE dataset [Gross et al., 2010] in Figs. A.2 and A.3.

[1] Source: https://unsplash.com/s/photos/face

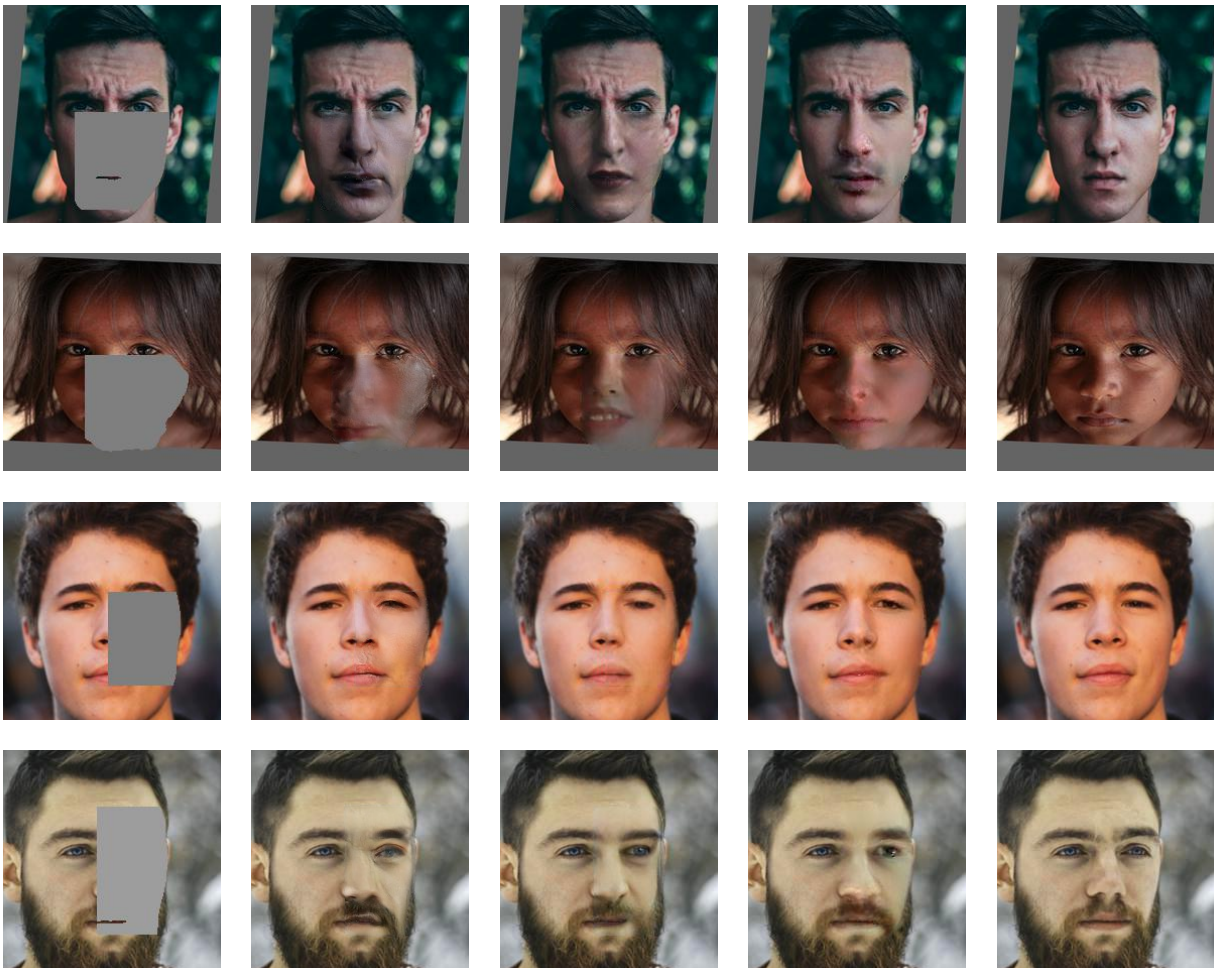


Figure A.1 Qualitative evaluation (of generalization performance) on the Internet downloaded images.

Figure A.2 Qualitative evaluation of Diverse3DFace *vs.* baselines DeepFillv2 [Yu et al., 2019] and PIC [Zheng et al., 2019a] on the pose-varying MultiPIE:Pose split [Gross et al., 2010]. While the baselines tend to generate blurred and deformed faces in extreme poses, Diverse3DFace is pose-robust and generates more accurate completions across a range of pose.

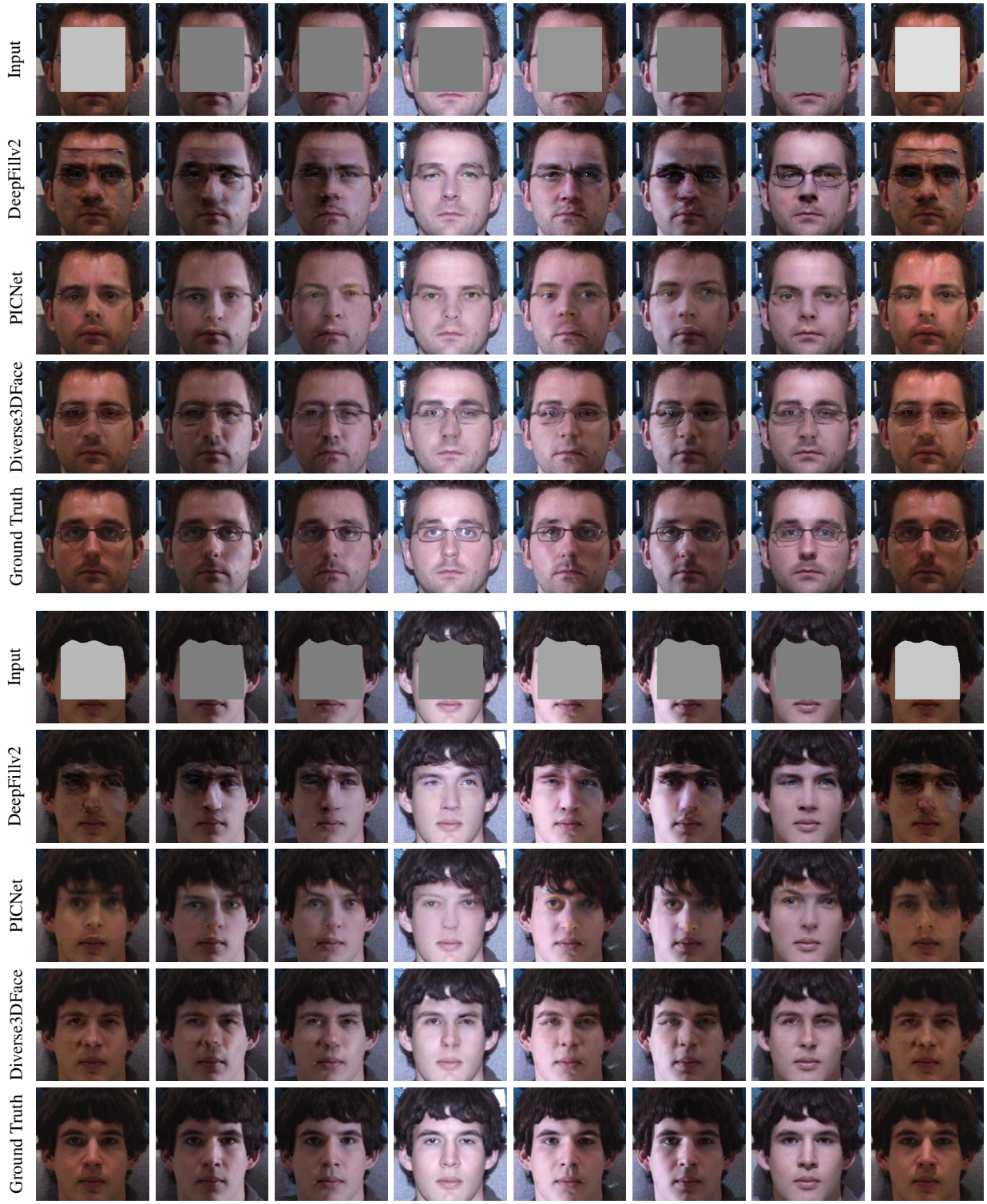


Figure A.3 Qualitative evaluation of Diverse3DFace *vs.*the baselines DeepFillv2 [Yu et al., 2019] and PIC [Zheng et al., 2019a] on the illumination varying MultiPIE:Illu split [Gross et al., 2010]. While the baselines tend to generate artifacts in extreme illuminations, Diverse3DFace generates completions that look geometrically accurate and preserve the illumination contrast.

A.3 Implementation Details

In this section, we provide further implementation details on 3DFaceFill. In sub-section A.3.1, we give detailed network architectures for the modules used in 3DFaceFill. In sub-section A.3.2, we provide details of the loss functions used to train the 3D factorization module. Lastly, we give full training details of the different components in sub-section A.3.3.

A.3.1 Network Architectures

We report the detailed network architectures for the 3DMM Encoder $\mathcal{E}$, the Albedo Decoder $\mathcal{G}_A$, the Sym-UNet module, the PyramidGAN discriminator and the Face Parser in Tables A.1 to A.5. Our network architectures for the 3DMM modules are based on the architectures used in [Tran and Liu, 2019] for the corresponding modules. Insipired by Miyato *et al*. [Miyato et al., 2018], we use spectral normalization in all our convolution layers. The abbreviated operators used are defined as follows:

- Conv$(c_{in}, c_{out}, k, s, p)$: 2D convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- Deconv$(c_{in}, c_{out}, k, s, p)$: 2D transposed convolution (deconvolution) with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- GN(n): Group normalization [Wu and He, 2018] with n groups
- ELU: Exponential linear unit [Clevert et al., 2015] activation, LReLU(α): Leaky ReLU [Maas et al., 2013] with a negative slope of α
- ResUnit$(c_{in}, c_{out}, k, s, p)$: Residual unit [He et al., 2016] with c_{in} input channels, c_{out} output channels, kernel size k, stride s, padding p with group normalization [Wu and He, 2018] and ELU activation [Maas et al., 2013]
- SigGNConv$(c_{in}, c_{out}, k, s, p)$: 2D convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p followed by group normalization [Wu and He, 2018] and sigmoid activation

- SigGNDeconv$(c_{in}, c_{out}, k, s, p)$: 2D transposed convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p followed by group normalization [Wu and He, 2018] and sigmoid activation

- SpectralConv$(c_{in}, c_{out}, k, s, p)$: 2D convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s, padding p and spectral normalization [Miyato et al., 2018]

- Upsample(s_h, s_c): Upsamples height by s_h and width by s_w using nearest neighbour interpolation.

A.3.2 3DMM Module Losses

The 3DMM module is trained using a combination of supervised, reconstruction and regularization losses:

$$\mathcal{L}_{\text{3DMM}} = \lambda_{\text{sup}}\mathcal{L}_{\text{sup}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}, \tag{A.1}$$

where, $\mathcal{L}_{\text{sup}} = \lambda_S\mathcal{L}(\mathbf{S}_{\text{gt}}, \tilde{\mathbf{S}}) + \lambda_\theta\mathcal{L}_\theta + \lambda_T\mathcal{L}(\mathbf{T}^{\text{uv}}_{\text{gt}}, \tilde{\mathbf{T}}^{\text{uv}}) + \lambda_{\text{lmark}}\mathcal{L}_{\text{lmark}}$ use the groundtruth shape $\mathbf{S}_{\text{gt}}$, pose $\boldsymbol{\theta}_{\text{gt}}$, texture $\mathbf{T}^{\text{uv}}_{\text{gt}}$ and 2D landmarks when available, $\mathcal{L}_{\text{rec}}$ enforces similarity between the rendered and grountruth images, and $\mathcal{L}_{\text{reg}} = \lambda_{\text{3dsym}}\mathcal{L}_{\text{3dsym}} + \lambda_{\text{const}}\mathcal{L}_{\text{const}}$ are regularization losses to enforce bilateral symmetry of albedo and effective separation of shade and albedo. All loss coefficients λ's are set to have equal weightage for all the loss terms. We now define these losses:

- Shape loss is defined as:

$$\mathcal{L}(\mathbf{S}_{\text{gt}}, \tilde{\mathbf{S}}) = \mathbb{E}\left[\left||\mathbf{S}_{\text{gt}} - \tilde{\mathbf{S}}|\right|_2^2\right],$$

where $\mathbf{S}_{\text{gt}}$ and $\tilde{\mathbf{S}}$ are the groundtruth and predicted 3D shapes, respectively.

- Pose loss is defined as a combination of scale, translation and rotation losses:

$$\mathcal{L}_\theta = \lambda_s\mathbb{E}\left[(s_{\text{gt}} - \tilde{s})^2\right] + \lambda_t\mathbb{E}\left[||\mathbf{t}_{\text{gt}} - \tilde{\mathbf{t}}||_2^2\right] + \lambda_r\mathcal{L}_R,$$

where s represents scale, $\mathbf{t}$ represents the translation, and $\mathcal{L}_R = \mathbb{E}\left[||\text{quat}(R_{\text{gt}}) - \text{quat}(\tilde{R})||_2^2\right]$ is the rotation loss with R representing the rotation along the X, Y and Z axes and quat$(.)$ gives its quaternion representation.

3DMM Encoder	Output size
Image → SpectralConv(3, 32, 7, 2, 3) + GN(8) + ELU	112x112
SpectralConv(32, 64, 3, 1, 1) + GN(16) + ELU	112x112
SpectralConv(64, 64, 3, 2, 1) + GN(16) + ELU	56x56
SpectralConv(64, 96, 3, 1, 1) + GN(24) + ELU	56x56
SpectralConv(96, 128, 3, 1, 1) + GN(32) + ELU	56x56
SpectralConv(128, 128, 3, 2, 1) + GN(32) + ELU	28x28
SpectralConv(128, 196, 3, 1, 1) + GN(48) + ELU	28x28
SpectralConv(196, 256, 3, 1, 1) + GN(64) + ELU	28x28
SpectralConv(256, 256, 3, 2, 1) + GN(64) + ELU	14x14
SpectralConv(256, 256, 3, 1, 1) + GN(64) + ELU	14x14
SpectralConv(256, 256, 3, 1, 1) + GN(64) + ELU	14x14
SpectralConv(256, 512, 3, 2, 1) + GN(128) + ELU	7x7
SpectralConv(512, 512, 3, 1, 1) + GN(128) + ELU → *feats*	7x7
feats → SpectralConv(512, 160, 3, 1, 1) + GN(40) + ELU	7x7
AvgPool(7,7)	1x1
Linear(160, 6) + Tanh → *Pose*	
feats → SpectralConv(512, 160, 3, 1, 1) + GN(40) + ELU	7x7
AvgPool(7,7)	1x1
Linear(160, 27) → *Illumination*	
feats → SpectralConv(512, 512, 3, 1, 1) + GN(128) + ELU	7x7
SpectralConv(512, 512, 3, 1, 1) + GN(128) + ELU	7x7
AvgPool(7,7)	1x1
Linear(512, 199+29) → *199 Shape + 29 Expression coefficients*	
feats → SpectralConv(512, 512, 3, 1, 1) + GN(128) + ELU	7x7
AvgPool(7,7) → *Albedo features*	1x1
Model Complexity	17.4M

Table A.1 Network architecture of the 3DMM Encoder $\mathcal{E}$. The $Pose_1$ corresponds to the scale, $Pose_{2:4}$ correspond to the yaw, roll and pitch angles normalized by $\pi/2$ and $Pose_{5:6}$ correspond to the X and Y translations normalized by the input image size.

Albedo Decoder	Output size
Albedo features $\rightarrow$ Upsample(3,4)	3x4
SpectralConv(512, 512, 3, 1, 1) + GN(128) + ELU	3x4
SpectralConv(512, 256, 3, 1, 1) + GN(64) + ELU	3x4
Upsample(2,2)	6x8
SpectralConv(256, 256, 3, 1, 1) + GN(64) + ELU	6x8
SpectralConv(256, 128, 3, 1, 1) + GN(32) + ELU	6x8
SpectralConv(128, 128, 3, 1, 1) + GN(32) + ELU	6x8
Upsample(2,2)	12x16
SpectralConv(128, 160, 3, 1, 1) + GN(40) + ELU	12x16
SpectralConv(160, 96, 3, 1, 1) + GN(32) + ELU	12x16
SpectralConv(96, 128, 3, 1, 1) + GN(32) + ELU	12x16
Upsample(2,2)	24x32
SpectralConv(128, 128, 3, 1, 1) + GN(32) + ELU	24x32
SpectralConv(128, 64, 3, 1, 1) + GN(16) + ELU	24x32
SpectralConv(64, 96, 3, 1, 1) + GN(24) + ELU	24x32
Upsample(2,2)	48x64
SpectralConv(96, 96, 3, 1, 1) + GN(32) + ELU	48x64
SpectralConv(96, 64, 3, 1, 1) + GN(16) + ELU	48x64
SpectralConv(64, 64, 3, 1, 1) + GN(16) + ELU	48x64
Upsample(2,2)	96x128
SpectralConv(64, 64, 3, 1, 1) + GN(16) + ELU	96x128
SpectralConv(64, 32, 3, 1, 1) + GN(8) + ELU	96x128
SpectralConv(32, 32, 3, 1, 1) + GN(8) + ELU	96x128
Upsample(2,2)	192x256
SpectralConv(32, 32, 3, 1, 1) + GN(8) + ELU	192x256
SpectralConv(32, 16, 3, 1, 1) + GN(4) + ELU	192x256
SpectralConv(16, 16, 3, 1, 1) + GN(4) + ELU	192x256
Conv(16, 3, 1, 1, 0) + Tanh $\rightarrow$ *Albedo*	
Model Complexity	5.54M

Table A.2 Network architecture of the Albedo Decoder $\mathcal{D}_A$ that decodes the 512 dimensional Albedo features from the 3DMM Encoder $\mathcal{E}$ into $3 \times 192 \times 256$ dimensional Albedo representation in the UV space.

Input	Layer	Output
X	ResUnit(4, 32, 3, 2, 1)	$f1$
X	SigGNConv(4, 32, 3, 2, 1)	$g1$
$hflip(X)$	ResUnit(4, 32, 3, 2, 1)	$f1'$
$hflip(X)$	SigGNConv(4, 32, 3, 2, 1)	$g1'$
$(f1 \odot g1, f1' \odot g1')$	ResUnit(64, 64, 3, 2, 1)	$f2$
$(f1 \odot g1, f1' \odot g1')$	SigGNConv(64, 64, 3, 2, 1)	$g2$
$f2 \odot g2$	ResUnit(64, 128, 3, 2, 1)	$f3$
$f2 \odot g2$	SigGNConv(64, 128, 3, 2, 1)	$g3$
$f3 \odot g3$	ResUnit(128, 256, 3, 2, 1)	$f4$
$f3 \odot g3$	SigGNConv(128, 256, 3, 2, 1)	$g4$
$f4 \odot g4$	ResUnit(256, 512, 3, 2, 1)	$f5$
$f4 \odot g4$	SigGNConv(256, 512, 3, 2, 1)	$g5$
$f5 \odot g5$	ResUnit(512, 256, 3, 1, 1)	$f5^1$
$f5 \odot g5$	SigGNConv(512, 256, 3, 1, 1)	$g5^1$
$f5^1 \odot g5^1$	Upsample(2,2)	$x4$
$(x4, f4 \odot g4)$	ResUnit(512, 128, 3, 1, 1)	$f4^1$
$f5^1 \odot g5^1$	SigGNDeconv(256, 128, 4, 2, 1)	$g4^1$
$f4^1 \odot g4^1$	Upsample(2,2)	$x3$
$(x3, f3 \odot g3)$	ResUnit(256, 64, 3, 1, 1)	$f3^1$
$f4^1 \odot g4^1$	SigGNDeconv(128, 64, 4, 2, 1)	$g3^1$
$f3^1 \odot g3^1$	Upsample(2,2)	$x2$
$(x2, f2 \odot g2)$	ResUnit(128, 64, 3, 1, 1)	$f2^1$
$f3^1 \odot g3^1$	SigGNDeconv(128, 64, 4, 2, 1)	$g2^1$
$f2^1 \odot g2^1$	Upsample(2,2)	$x1$
$(x1, f1 \odot g1)$	ResUnit(128, 64, 3, 1, 1)	$f1^1$
$f2^1 \odot g2^1$	SigGNDeconv(128, 64, 4, 2, 1)	$g1^1$
$f1^1 \odot g1^1$	Upsample(2,2)	$x0$
$x0$	ResUnit(64, 32, 3, 1, 1)	$f0^1$
$f1^1 \odot g1^1$	SigGNDeconv(64, 32, 4, 2, 1)	$g0^1$
$f0^1 \odot g0^1$	Conv(32, 4, 1, 1, 0)	$(\hat{\mathbf{A}}^{uv}, \sigma^{uv})$
Model Complexity		11.7M

Table A.3 Network architecture of the Albedo Inpainter $\mathcal{G}$ (Sym-UNet). The input to the network is the concatenation of the masked Albedo $\mathbf{A}_m^{uv}$ and the mask $\mathbf{M}^{uv}$ in the UV space $X = (\mathbf{A}_m^{uv}, \mathbf{M}^{uv})$. Outputs are the completed Albedo $\hat{\mathbf{A}}^{uv}$ and the uncertainty map σ^{uv}.

Input	Layer	Output
$\mathbf{I}_{gt}/\hat{\mathbf{I}}$	SpectralConv(3, 32, 4, 2, 1) + GN(8) + LReLU(.2)	$x0$
$x0$	SpectralConv(32, 64, 4, 2, 1) + GN(16) + LReLU(.2)	$x1$
$x1$	SpectralConv(64, 1, 1, 1, 0)	$out1$
$x1$	SpectralConv(64, 128, 4, 2, 1) + GN(32) + LReLU(.2)	$x2$
$x2$	SpectralConv(128, 1, 1, 1, 0)	$out2$
$x2$	SpectralConv(128, 256, 4, 2, 1) + GN(64) + LReLU(.2)	$x3$
$x3$	SpectralConv(256, 1, 1, 1, 0)	$out3$
$x3$	SpectralConv(256, 512, 4, 2, 1) + GN(128) + LReLU(.2)	$x4$
$x4$	SpectralConv(512, 1, 1, 1, 0)	$out4$
	Model Complexity	2.79M

Table A.4 Network architecture of the PyramidGAN discriminator $\mathcal{D}$.

Input	Layer	Output
Image	ResUnit(3, 32, 3, 1, 1)	$x1$
$x1$	ResUnit(32, 64, 3, 2, 1)	$x2$
$x2$	ResUnit(64, 128, 3, 2, 1)	$x3$
$x3$	ResUnit(128, 256, 3, 2, 1)	$x4$
$x4$	ResUnit(256, 256, 3, 2, 1)	$x5$
$x5$	ResUnit(256, 256, 3, 2, 1)	$x6$
$x6$	Upsample(2,2)	$x5^1$
$(x5^1, x5)$	ResUnit(512, 256, 3, 1, 1)	$x5^2$
$x5^2$	Upsample(2,2)	$x4^1$
$(x4^1, x4)$	ResUnit(512, 128, 3, 1, 1)	$x4^2$
$x4^2$	Upsample(2,2)	$x3^1$
$(x3^1, x3)$	ResUnit(256, 64, 3, 1, 1)	$x3^2$
$x3^2$	Upsample(2,2)	$x2^1$
$(x2^1, x2)$	ResUnit(128, 32, 3, 1, 1)	$x2^2$
$x2^2$	Upsample(2,2)	$x1^1$
$(x1^1, x1)$	ResUnit(64, 32, 3, 1, 1)	$x1^2$
$x1^2$	Conv(32, 3, 1, 1, 0) + Softmax2d	$(\mathbf{M}_f, \mathbf{M}_o, \mathbf{M}_b)$
	Model Complexity	7.18M

Table A.5 Network architecture of the face parser. (x, y) represents the concatenation of tensors x and y along the channel dimension. The output of the network consist of a face mask $\mathbf{M}_f$, an occlusion mask $\mathbf{M}_o$ and a background mask $\mathbf{M}_b$.

- Texture loss is defined as:

$$\mathcal{L}(\mathbf{T}_{\text{gt}}^{\text{uv}}, \tilde{\mathbf{T}}^{\text{uv}}) = \mathbb{E}\left[||\mathbf{T}_{\text{gt}}^{\text{uv}} - \tilde{\mathbf{T}}^{\text{uv}}||_2^2\right],$$

where $\mathbf{T}^{\text{uv}}$ is the texture represented in UV space.

- Landmark loss is defined as:

$$\mathcal{L}_{\text{lmark}} = \left\| \mathbf{M}(\grave{}) * \begin{bmatrix} \mathbf{S}(:, \mathbf{d}) \\ \mathbf{1} \end{bmatrix} - \mathbf{U} \right\|_2^2,$$

where $\mathbf{M}$ is the camera projection matrix obtained from the pose $\boldsymbol{\theta}$, $\mathbf{d}$ selects 68 indices corresponding to sparse 2D landmarks on the 3D face mesh $\mathbf{S}$ and $\mathbf{U} \in \mathbb{R}^{68\times 2}$ are the groundtruth locations of 2D facial landmarks.

- Reconstruction loss is defined as:

$$\mathcal{L}_{\text{rec}} = ||(\mathbf{I}_{\text{gt}} - \mathbf{I}_{\text{rec}}) \odot \mathbf{M}_f||_2^2,$$

where $\mathbf{I}_{\text{gt}}$ and $\mathbf{I}_{\text{rec}}$ are the original and the rendered images, respectively and $\mathbf{M}_f$ is the face mask.

- Albedo symmetry loss is defined as:

$$L_{\text{3dsym}}(\mathbf{A}) = \|\mathbf{A}^{\text{uv}} - \text{hflip}(\mathbf{A}^{\text{uv}})\|_1,$$

where $\mathbf{A}^{\text{uv}}$ is the UV representation of albedo and hflip() is the horizontal image flipping operation.

- Albedo constancy loss is defined as:

$$L_{\text{const}}(\mathbf{A}) = \sum_{\mathbf{v}_j^{\text{uv}} \in \mathcal{N}_i} \omega(\mathbf{v}_i^{\text{uv}}, \mathbf{v}_j^{\text{uv}}) \|\mathbf{A}^{\text{uv}}(\mathbf{v}_i^{\text{uv}}) - \mathbf{A}^{\text{uv}}(\mathbf{v}_j^{\text{uv}})\|_2^p,$$

where $\mathcal{N}_i$ denotes the 4-neighborhood around $\mathbf{v}_i^{\text{uv}}$ and the weight $\omega(\mathbf{v}_i^{\text{uv}}, \mathbf{v}_j^{\text{uv}}) = \exp(-\alpha \|c(\mathbf{v}_i^{\text{uv}}) - c(\mathbf{v}_j^{\text{uv}})\|)$ enforce that pixels with similar chromaticity should have similar albedo.

A.3.3 Training Details

3DMM Module: We train the 3DMM module in two stages. First, we train it using the 300W-3D dataset [Zhu et al., 2016], which has ground-truth shape, pose, texture and landmark annotations,

for 100k iterations in a supervised way. Then, we further train it on the CelebA dataset [Liu et al., 2015] with 1/10th of the original learning rate for further 30k iterations in an unsupervised way, whereby we use only the reconstruction loss, 2D landmark loss and the regularization losses. During this stage, we use landmark detections from HRNet [Wang et al., 2020] as groundtruth for the landmark loss. To make the 3DMM encoder robust to partial face images, we introduce artificial occlusions in the training images using random rectangular masks of varying sizes and locations. In addition, we also use random horizontal flipping as a data augmentation. During inference, occlusions are removed from the input image using the occlusion mask and passed through the 3DMM encoder to obtain occlusion-robust factorization.

Albedo Inpainting Module: The albedo inpainting module is trained on the CelebA dataset [Liu et al., 2015] for 30k iterations. To obtain the UV representations of the partial albedo and the mask, we re-project the 3D mesh obtained from the pretrained 3DMM module on the partial image and mask, respectively as shown in Fig. 3.2. On the GAN loss Eq. (3.3), we update the inpainter $\mathcal{G}$ and the discriminator $\mathcal{D}$ alternatively using a ratio of 1:1. On all the other completion losses, we update the inpainter $\mathcal{G}$ continuously. Other than the random face masks, we use random horizontal flipping as the only data augmentation to train the albedo inpainter.

Face Segmentation Module: Since our method inpaints only the facial region in the UV domain, we restrict the image masks to lie on the face region too. For this, we train a UNet [Ronneberger et al., 2015] based face segmentation model that separates the face region from the background, hair and inner mouth. The face segmenter predicts segmentation masks for (a) the face, (b) hair and other occlusions and (c) the background. We train the face segmentation module on the CelebAMask-HQ dataset [Lee et al., 2020] for a total of 50k iterations using the ground-truth annotations provided by the dataset. We use Focal loss [Lin et al., 2017] to train this module.

For all the modules, except the discriminator $\mathcal{D}$, we use the Adam optimizer with an initial learning rate of 10^{-4} and a step-decay of 0.98 per epoch, while for the PyramidGAN discriminator, we use an initial learning rate of 3×10^{-4}. The input images are first aligned to 256×256 using the method suggested in [Lee et al., 2020], which is the alignment used in the CelebA-HQ dataset.

For training, we randomly crop the images to a size of 224×224 while during inference we use central crop. The full training takes 2 days on an Intel Xeon E5-2650 machine with two NVIDIA RTX 2080 GPUs, while inference takes 0.1 sec per image on a single GPU.

APPENDIX B COLA-SDF

B.1 Attribute Transfer

We show additional source-to-target attribute transfer results, including shape transfer in Fig. B.1, texture transfer (transfer of both albedo and illumination) in Fig. B.2, and hair/background transfer in Fig. B.3. In Fig. B.3, we again observe that while the hair geometry and style is mainly controlled by the hair/background code, its appearance is partly controlled by the albedo and illumination codes. These results show CoLa-SDF's ability to transfer one attribute while keeping the rest intact and demonstrate the attribute disentangled latent space learned by CoLa-SDF.

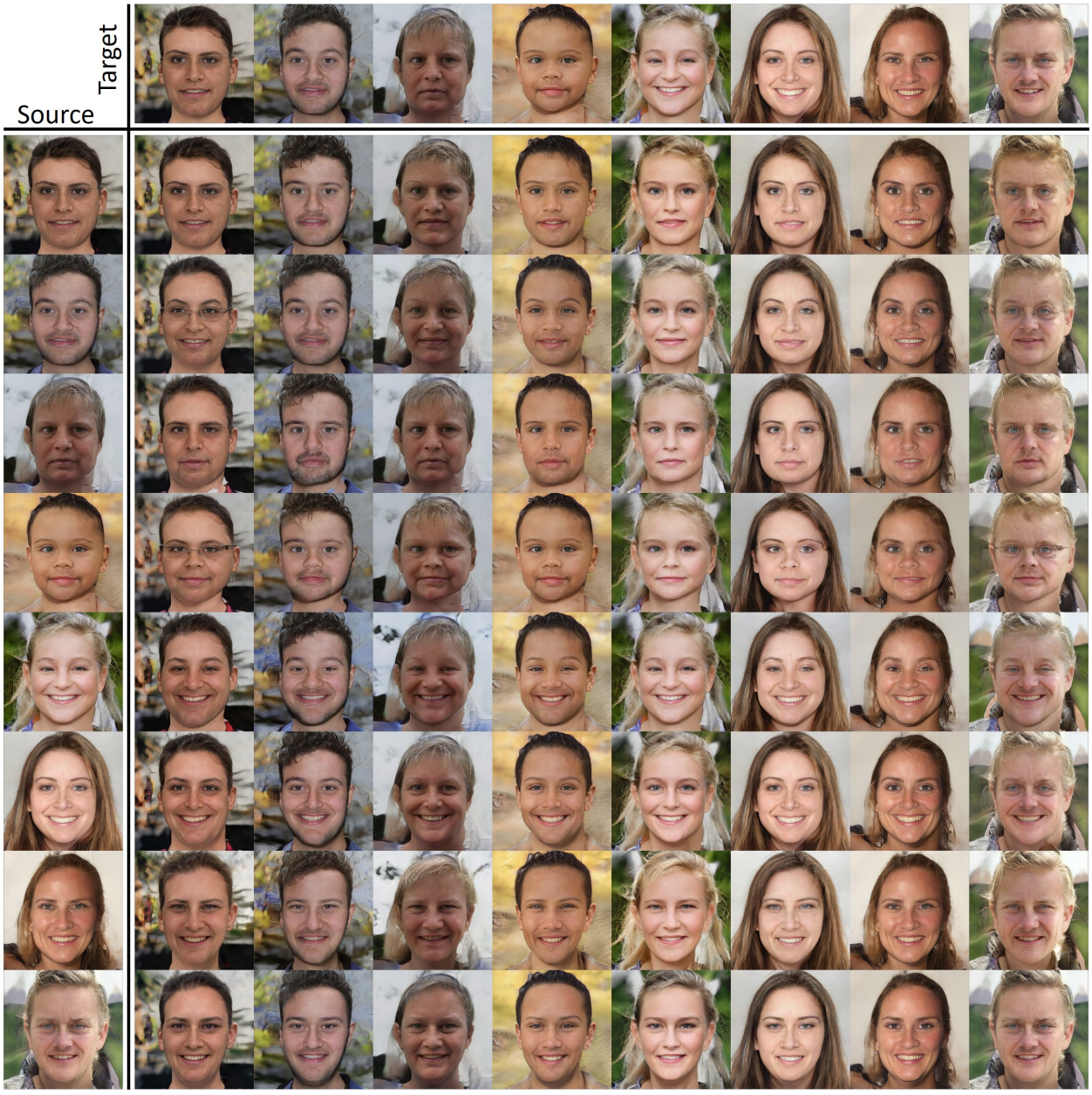


Figure B.1 Further shape transfer results using CoLa-SDF.

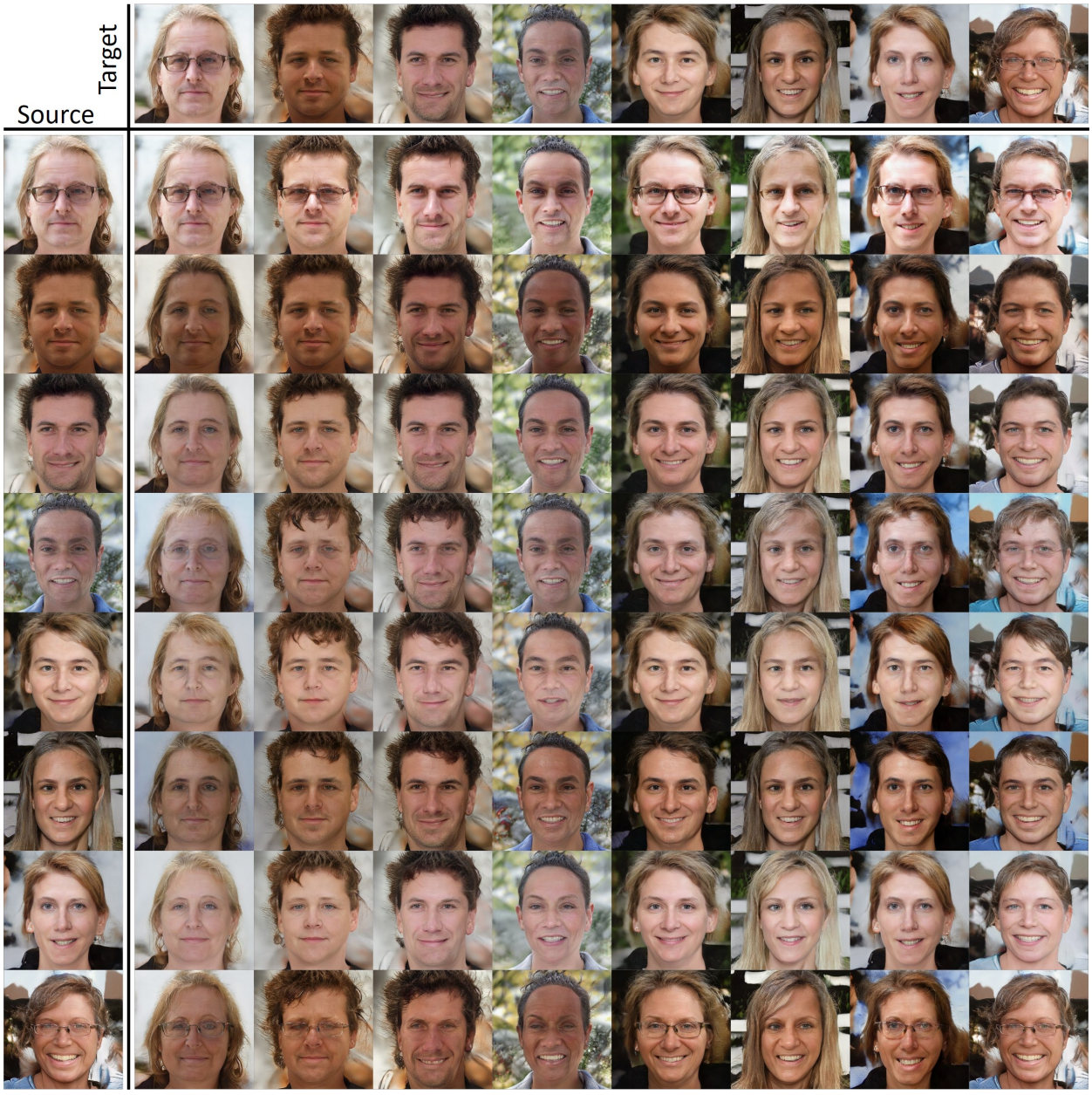


Figure B.2 Further texture (albedo + illumination) transfer results using CoLa-SDF.

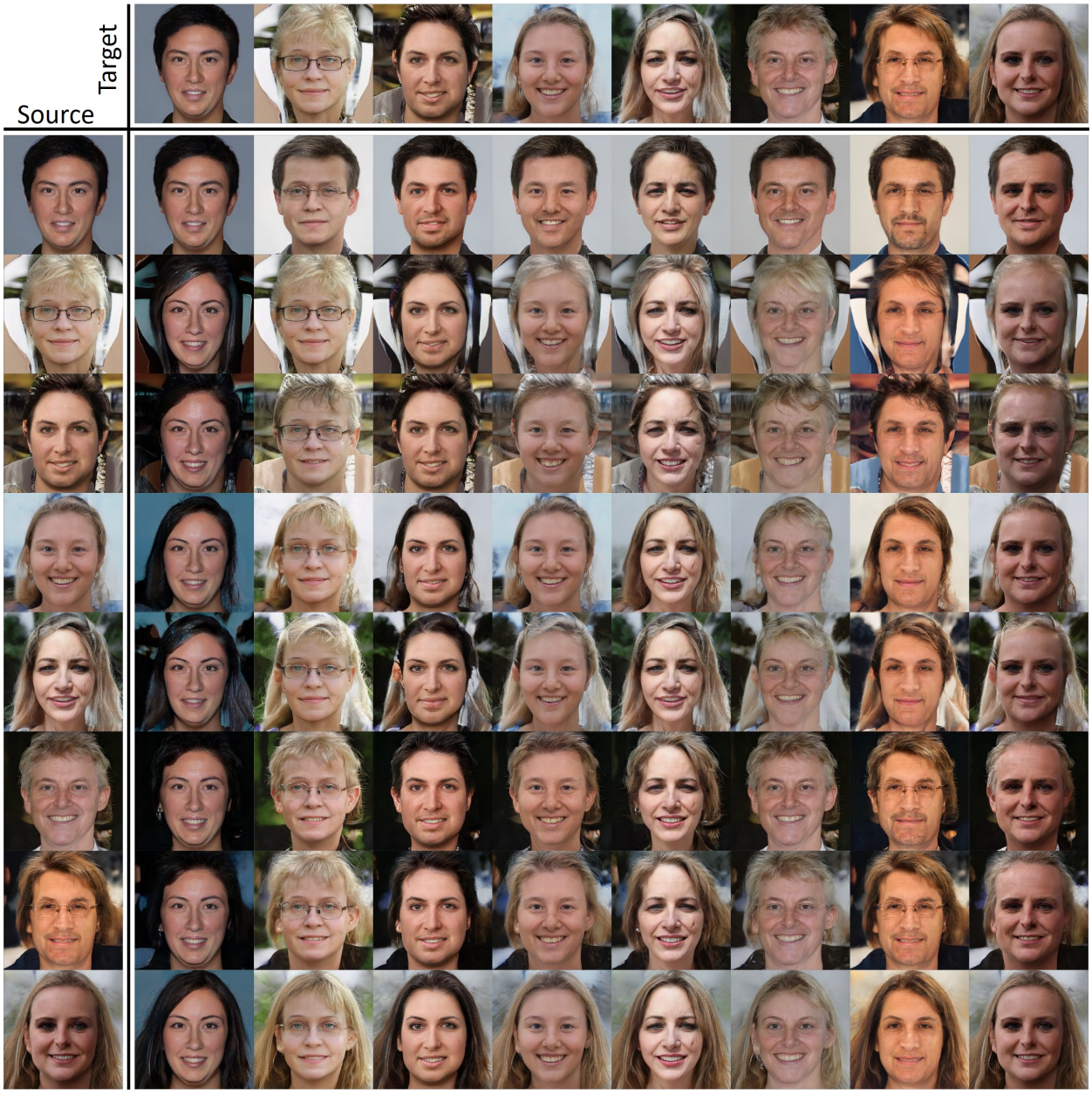


Figure B.3 Further hair/background transfer results using CoLa-SDF.

APPENDIX C DIVERSE3DFACE

C.1 Implementation Details

C.1.1 Optimization

We use the *PyTorch* library to implement our approach. In our experiments, we found that the SGD optimizer, with a learning rate of 5×10^{-3} gives the best results as compared to the Adam and RMSprop optimizers. For photometric fitting, we used the texture model provided by https://flame.is.tue.mpg.de/index.htmlFLAME. We run the fitting stage (Algorithm 1) for $n_{\text{iter}} = 2000$ iterations and the diversity stage (Algorithm 2) for $n_{\text{comp}} = 300$ iterations. In Algorithm 1, we set the loss weights as follows: $\lambda_1^f = 5, \lambda_2^f = 16, \lambda_3^f = 10^{-3}$. During the diversifying shape completion stage (Algorithm 2), we set $\lambda_1 = 1000, \lambda_2 = 500, \lambda_3 = 0.025$. Further, we found that using a slightly smaller learning rate for the eyeball components while fitting the global+local model gives better results. For these components, we set the learning rate to be 0.5 times that of the other components.

C.1.2 Mesh-VAE

The Mesh-VAE model is based on the fully convolutional mesh autoencoder (Meshconv) architecture proposed by Zhou *et al*. [Zhou et al., 2020b]. Meshconv [Zhou et al., 2020b] uses spatially varying convolutional kernels for different mesh vertices to account for the irregular structure of a 3D mesh. The spatially varying kernels are sampled from the span of a shared weight basis, using learned per-vertex coefficients. In addition, Meshconv defines pooling and unpooling operations on a 3D mesh by performing feature aggregation Monte Carlo sampling [Zhou et al., 2020b].

We trained the Mesh-VAE with FLAME [Li et al., 2017a] registered groundtruth scans provided in the CoMA [Ranjan et al., 2018] and D3DFACS [Cosker et al., 2011] datasets. We perturbed the input meshes with uniformly sampled rectangular masks (in XY) within a range around the mesh center, while gradually increasing the size of the mask per training epoch until it covered $\sim$40% of the vertices. We detail the network architecture for the Mesh-VAE in Tabs. C.1 and C.2.

The abbreviated operators used are defined as follows:

Input	Layer	Output size	Output
5023×3 **Mesh**	$\rightarrow$ vcDownConv($in_c = 3, out_c = 32, s = 2, r = 43, M = 17$) + vcDownRes(2)	1367×32	
	vcDownConv($in_c = 32, out_c = 64, s = 1, r = 27, M = 17$) + vcDownRes(1)	1367×64	
	vcDownConv($in_c = 64, out_c = 128, s = 2, r = 54, M = 17$) + vcDownRes(2)	270×128	
	vcDownConv($in_c = 128, out_c = 256, s = 1, r = 25, M = 17$) + vcDownRes(1)	270×256	
	vcDownConv($in_c = 256, out_c = 512, s = 2, r = 81, M = 17$) + vcDownRes(2)	45×512	
	vcDownConv($in_c = 512, out_c = 1024, s = 1, r = 27, M = 17$) + vcDownRes(1)	45×1024	*feats*
feats	vcDownConv($in_c = 1024, out_c = 64, s = 2, r = 37, M = 17$) + vcDownRes(2)	10×64	$\boldsymbol{\mu}$
feats	vcDownConv($in_c = 1024, out_c = 64, s = 2, r = 37, M = 17$) + vcDownRes(2)	10×64	$\log \boldsymbol{\sigma}^2$
Model Complexity	9M		

Table C.1 Network architecture of the Mesh-VAE Encoder $\mathcal{E}_{\text{mesh}}$.

Input	Layer	Output size	Output
10×64 **z**	vcUpConv($in_c = 64, out_c = 1024, s = 2, r = 8, M = 17$) + vcUpRes(2)	45×1024	
	vcUpConv($in_c = 1024, out_c = 512, s = 1, r = 27, M = 17$) + vcUpRes(1)	45×512	
	vcUpConv($in_c = 512, out_c = 256, s = 2, r = 16, M = 17$) + vcUpRes(2)	270×256	
	vcUpConv($in_c = 256, out_c = 128, s = 1, r = 25, M = 17$) + vcUpRes(1)	270×128	
	vcUpConv($in_c = 128, out_c = 64, s = 2, r = 12, M = 17$) + vcUpRes(2)	1367×64	
	vcUpConv($in_c = 64, out_c = 32, s = 1, r = 27, M = 17$) + vcUpRes(1)	1367×32	
	vcUpConv($in_c = 32, out_c = 3, s = 2, r = 24, M = 17$) + vcUpRes(2)	5023×3	**Output**
Model Complexity	8M		

Table C.2 Network architecture of the Mesh-VAE Decoder $\mathcal{D}_{\text{mesh}}$.

- vcDownConv(in_c, out_c, s, r, M) + vcDownRes(s): Downward residual block (as defined in Meshconv [Zhou et al., 2020b]), with in_c input channels, out_c output channels, s stride, r kernel radius and M number of shared weight bases. The output is activated with ELU [Clevert et al., 2015] activation.

- vcUpConv(in_c, out_c, s, r, M) + vcUpRes(s): Upward residual block (as defined in Meshconv [Zhou et al., 2020b]), with in_c input channels, out_c output channels, s stride, r kernel radius and M number of shared weight bases. The output is activated with ELU [Clevert et al., 2015] activation.