Inverting Generative Adversarial Renderer for Face Reconstruction

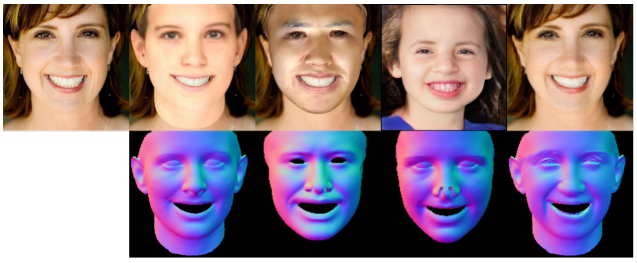
Jingtan Piao[1], Keqiang Sun[1], Quan Wang[2,3], Kwan-Yee Lin[1,2*], Hongsheng Li[1,4*]
[1]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[2]SenseTime Research and Tetras.AI [3]Shanghai AI Laboratory [4]School of CST, Xidian University
{1155116308, kqsun}@link.cuhk.edu.hk, {wangquan, linjunyi}@sensetime.com, hsli@ee.cuhk.edu.hk

Abstract

Given a monocular face image as input, 3D face geometry reconstruction aims to recover a corresponding 3D face mesh. Recently, both optimization-based and learning-based face reconstruction methods have taken advantage of the emerging differentiable renderer and shown promising results. However, the differentiable renderer, mainly based on graphics rules, simplifies the realistic mechanism of the illumination, reflection, etc.*, of the real world, thus cannot produce realistic images. This brings a lot of domain-shift noise to the optimization or training process. In this work, we introduce a novel Generative Adversarial Renderer (GAR) and propose to tailor its inverted version to the general fitting pipeline, to tackle the above problem. Specifically, the carefully designed neural renderer takes a face normal map and a latent code representing other factors as inputs and renders a realistic face image. Since the GAR learns to model the complicated real-world image, instead of relying on the simplified graphics rules, it is capable of producing realistic images, which essentially inhibits the domain-shift noise in training and optimization. Equipped with the elaborated GAR, we further proposed a novel approach to predict 3D face parameters, in which we first obtain fine initial parameters via Renderer Inverting and then refine it with gradient-based optimizers. Extensive experiments have been conducted to demonstrate the effectiveness of the proposed generative adversarial renderer and the novel optimization-based face reconstruction framework. Our method achieves state-of-the-art performances on multiple face reconstruction datasets.*

1. Introduction

Faithfully recovering the 3D shapes of human faces from unconstrained 2D images is a challenging task and has numerous applications such as face recognition and face animation [45, 48]. State-of-the-art 3D face reconstruction methods can be generally categorized into two groups, learning-based methods and optimization-based methods.



(a) Input (b) 3DDFA (c) GANFIT (d) DFG (e) Ours

Figure 1. Comparisons with state-of-the-art face renderers. On the second row are input geometry and the first row are corresponding rendered images. Output of (b) [50] and (c) [13] are not realistic, since they use graphics-based renderers. And there may exists inconsistency between the input and the rendered image in (d) [10]. Our method faithfully renders realistic images consistent with the input geometry, as shown in (e).

The deep learning-based methods [50, 8, 14, 11] usually take place in a regression manner, which takes facial images as inputs and learn to regress the corresponding 3DMM parameters. However, these methods usually require large amounts of labeled data, while the ground truth 3DMM parameters are rather difficult to acquire. Optimization-based methods [5, 22, 13, 49], on the other hand, generally treat the imaging of faces as a generative process [29], which takes a series of geometry coefficients (*e.g.*, albedo, texture, lighting, viewing angle, etc.) as inputs and outputs a rendered image according to certain graphics rules. The distances between the rendered images and the target images are minimized with an optimization framework. However, since the graphics rules generally employ simplified models to characterize the physical process of capturing face images, many details of the imaging process cannot be modeled, which introduces difficulties for the optimization of face reconstruction.

Recent developments of the differentiable renderers provide an efficient tool for both types of face reconstruction methods. Specifically, the regressed parameters in learning-based methods could be rendered to images, with which the photometric loss can be adopted for optimization. In this manner, as shown in [11], learning-based models may be

*K. Lin and H. Li are the co-corresponding authors.

trained without geometry ground truth of the input image. For the optimization-based methods, as introduced by [13], differentiable renderers introduce gradient-based optimization and allow adopting more complicated losses and stabilizes the training process.

However, differentiable renderers have two drawbacks. On the one hand, the differentiable renderers are created by handcrafted rendering rules and are generally not capable of producing realistic images. The domain gap between the rendered and real images hinders the optimization or the training process. On the other hand, the differentiable renderers are difficult to optimize as they can only back-propagate errors to local vertices. As shown in (b) and (c) of Figure 1, the rendered image is not realistic since they are using graphics-based renderers. Some methods [22, 25] modify the renderers to make them "more" differentiable and better converge to the optimum via optimization, whereas they are still utilizing the graphics-based rendering methods, hence the above two problems remain essential drawbacks of the differentiable renderer.

An intuitive solution is to replace the differentiable renderer with a neural renderer, an emerging method to employ a neural network to render an image corresponding with the given geometry and texture conditions. Actually, several types of neural renderers have been proposed and studied before. For instance, Deng *et al*. [10] proposed a neural renderer, which takes 3DMM parameters as inputs and generates a facial image. Nevertheless, the 3DMM parameters are too abstract for the control of the generative adversarial renderer. Therefore, the rendered images, although are more realistic and basically subject to the inputs, do not strictly condition on the 3DMM parameters. As shown in (d) of Figure 1, even though the input geometry parameters are close to the target person, the rendered image shows a large variation. Hence, it is not an ideal neural renderer for face reconstruction.

In this paper, we propose to adopt a novel conditional neural renderer, trained in a self-supervised manner, to replace the conventional graphics-based differentiable renderer, to tackle the aforementioned problems while maintaining the advantages of utilizing a renderer for training. The proposed conditional face neural renderer takes a face normal map as the geometry condition and a latent code vector to model other influencing factors. Since we hope the proposed renderer could facilitate the optimization of the face geometry, we decouple the normal map from the other condition factors so that the geometry could be better reconstructed via optimization of the normal map. To further enhance the controllability of the normal map upon the rendered images, a novel Normal Injection Module (NIM) is proposed, in which the normal map is used to modulate the convolution kernel by pixel-wise multiplication on each channel, to determine the geometry. On the other hand, the decoupled latent code contains detailed information about the facial textures, which are also significant in reconstructing the image faithfully. With a novel normal consistency loss, the whole neural renderer is trained in a self-supervised manner without any labeled data. As shown in (e) of Figure 1, the proposed GAR could faithfully render a realistic face image, according to the input geometry map.

After the neural renderer is trained, it takes the place of the differentiable renderer in the optimization-based face geometry reconstruction pipeline, in which the deviation between the given image and the rendered image is minimized and the geometry corresponding to the normal map is optimized.

Even with the proposed neural renderer, direct optimization with random initialization still struggles to recover the optimal 3D face shape. We further proposed a novel approach to predict 3D face parameters, in which we first predict a set of good initial 3D parameters by a separate neural network and then refine them with a gradient-based optimizer. Inspired by the latest GAN inverting technique [4], we train a regression network to predict a good initialization of the latent code for inverting the neural renderer to robustly recover the conditioning face normal map. The optimal face normal maps and subsequently the corresponding face shapes can then be obtained via iterative gradient-based optimization.

The proposed optimization algorithm has two unique advantages. 1) The optimization process is more stable because the "fully" differentiable neural renderer has larger receptive fields and can achieve more accurate image reconstruction. 2) With the proposed initialization-prediction network, the neural renderer can be easier inverted to convergence and achieve better accuracy on face reconstruction.

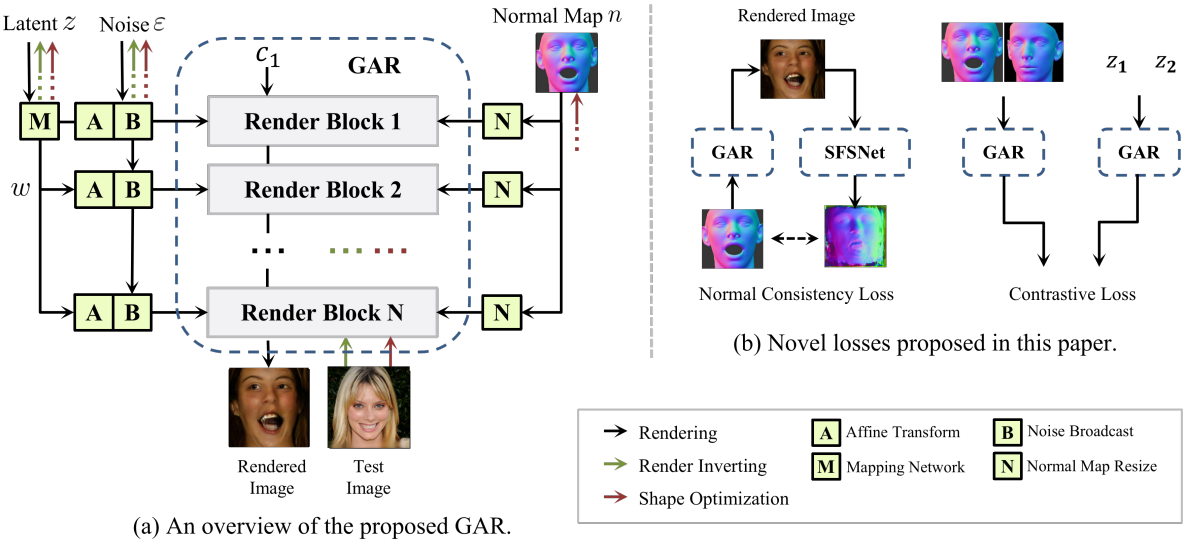
In summary, the main contributions of the proposed method are three-fold:

- To the best of our knowledge, we are the first to employ a conditional neural renderer, instead of a graphics-based differentiable renderer, to facilitate the face reconstruction.
- We propose a novel normal-conditioning neural renderer that can produce vivid face images conditioned on the input normal map and a latent code.
- We propose a face reconstruction algorithm based on the novel neural renderer, and achieve state-of-the-art performance on multiple face reconstruction datasets.

2. Related Work

2.1. 3D Face Modeling

Face modeling aims at using mathematical formulas to generate locations of vertices of a face mesh. Since the in-



(a) An overview of the proposed GAR.

(b) Novel losses proposed in this paper.

Figure 2. The architecture of the proposed method. (a) The proposed Generative Adversarial Renderer (GAR) is composed of a series of Render Blocks. The latent code z, mainly encoding non-geometry information, is transformed to the normalization parameters w through a mapping network M. The noise code ε is broadcast to each block with B. And the normal map is resized to certain resolutions with N. (b) In training, we introduced a novel Normal Consistency Loss to enhance the controllability of the input normal map, and a Contrastive Loss to decouple the normal n and the latent code z.

troduction of the Basel Face Model [6] that used a linear combination of Gaussian distributed coefficients of a face, multiple methods have been proposed, including blender shapes [9], skeleton animations [24], B-splines [28], deformation based method [18]. Nonlinear models have also been proposed [43, 42]. However, all of the face models can only formulate the facial regions and lack the modeling of the face accessories and hairs. Some efforts have been made to model various face attributes, including hair [16], mouth [30], back of the head [33, 23], *etc*. However, they are generally more complicated and require much more computation to fit or render a face image.

2.2. 3D Face Reconstruction

3D face reconstruction is the inverse process of recovering the face shapes from a monocular image. The methods can be generally categorized into two types, optimization-based methods and learning-based methods. The optimization-based methods generally try to invert the rendering or imaging process of face images by optimizing a cost function for each input image [46, 33]. Various loss functions were explored, including the re-projection loss of the detected 2D landmarks from the rendered 3D mesh [5], the photometric loss between the rendered image by a differentiable renderer and the original image [22], etc. [13] introduces a pretrained generative adversarial network to fit the texture UV map. And thanks to the differential renderers, [13] adopts photo-metric loss, as well as recognition loss to further enhance the texture and geometry quality. [49] proposes a ReDA Rasterizer for more soft and realistic rendering, and a free-form deformation layer with as-rigid-as-possible constraint to reconstruct an accurate face model.

Recently, deep learning-based methods have presented promising performance. Zhu *et al*. [50] proposed to directly regress the face parameters from the input face image. Starting from the cascaded face-parameter regressor [8], there are methods focusing on designing supervisions on representing the final reconstructed mesh. Video-based methods introduced additional constraints to regularize the reconstruction results. [14] assumed that the reconstructed face images from multiple frames of a video should maintain the same face identity and similar textures. Face recognition models and perceptual loss are adopted to minimize the differences between the feature maps of the multi-view images to better regularize the reconstructed face shapes.

2.3. Face Image Generation

Many 3D-based methods have been proposed to generate face images [12, 10]. Besides methods using a statistic face model to explicitly calculate the 3D mesh, face image auto-encoder has been popular since the introduction of generative adversarial network [15]. High-resolution images can be gradually generated from a low-resolution to high-resolution manner. StyleGANs [20, 21] introduced to model the latent code for image generation as the input Batch Normalization parameters. There are also methods that try to control the generated images with the input latent code [29] by adding a classifier to restrict the output

patterns. However, the controllable properties of the generated face images are only limited to the modification of single neurons [37]. Some feature disentanglement networks [1] for image generation have also been proposed. However, they cannot generate realistic images conditioned on 3D information. [32] tried to disentangle the process of 2D face image generation into 3D mid-level feature generation and 3D-to-2D feature projection and generation. Other than how to generate more realistic face images, given a trained GAN model, how to effectively invert the GAN to obtain the corresponding latent code is also of importance. [4] studied how to effectively invert a GAN model, since the inverting of the GAN model might also be stuck at local minima. A regression network is trained with the generator network predicting a good initialization for inverting the GAN from the input image to recover the corresponding latent code.

3. Method

The goal of this work is to reconstruct its corresponding face geometry parameters from a single image.

Given an image, 1) the latent code and noise will be initialized via GAR Inverting network, and 2) 3DMM parameters will be initialized with the fitting method. With parameters initialized from 1) and 2), we then finetune all parameters and latent codes by back-propagation.

The key of the proposed algorithm is a Generative Adversarial Renderer G, which is trained to generate realistic face images conditioned on the input face normal map and a latent code. G is trained in a self-supervised manner, and no labeled data is required. The renderer G is fixed after training. Given an unseen face image, a renderer inverting network R is trained to predict a good initialization for the latent code, based on which, a gradient-based optimizer can effectively recover the face geometry parameters.

3.1. Generative Adversarial Renderer

In this section, we introduce the proposed Generative Adversarial Renderer G, which takes in a normal map n and latent code z and outputs a corresponding rendered image I_{out}.

Architecture. The proposed Generative Adversarial Renderer G is composed of a series of Render Blocks, based on StyleGan v2 [21], as shown in Figure 2 (a). Each block, corresponding to a certain resolution, contains style-varying convolutions, modulated by a latent code w mapped from an input latent code z. The feature map after the convolution is then modulated by a normal map n, which can be generated from 3DMM face models with different shape α, expression β, and pose θ parameters. See Figure 3 for details.

The latent code z, which encodes factors of a face image other than its normal, is transformed to the normalization parameters w through a mapping network M for modulating kernel parameters of convolution in each block of the network. The modulation and demodulation of the style-varying kernel parameters k by the latent code z is defined as

$$k'_{cij} = w_c k_{cij} / \sqrt{\sum_{c,j} (w_c k_{cij})^2 + \epsilon}, \tag{1}$$

$$w = M(z), \tag{2}$$

where k_{cij} denotes the initial kernel parameter at spatial position (i, j) of the c-th channel, k'_{cij} denotes the kernel parameter after modulation, w_c is the modulation parameter for the c-th instance channel, predicted from the latent code z by an 8-layer MLP M, as represented in Equation 2. And ϵ here is used to avoid numerical division by zero.

The input feature map f_{cxy} is convoluted to f'_{lxy} with the modulated kernels

$$f'_{lxy} = \sum_{i,j} k^l_{cij} f_{c,x+i,y+j}, \tag{3}$$

where f'_{lxy} indicates the feature map pixel at (x, y) in the l-th channel.

The normal map is used to further modulate the feature map f' in the Normal Injection Module (NIM). However, instead of regularizing the channel dimension, the normal map is used for regularizing the spatial dimension:

$$f''_{lxy} = n_{xy} f'_{lxy}, \tag{4}$$

where f''_{lxy} is the feature maps after modulation by the input normal map, and n_{xy} denotes the injecting normal values from the facial normal map n from the spatial location (x, y).

The feature maps f'' are added with a learned bias b and a random Gaussian noise map ε before sent to the next block. The insight here, as discussed in the original StyleGAN [20], is that the small dimension of the latent code z cannot fully express all the details of a face image. The extra noise is therefore needed to properly model the extra information.

Loss Functions. Given a normal map n, a latent code z and a random noise ϵ, the neural renderer outputs a corresponding face image,

$$I_{\text{out}} = G(n, z, \varepsilon). \tag{5}$$

Besides the commonly used adversarial loss for encouraging image vividness, to regularize the results of G to match the input conditioning normal map, we propose a cycle normal consistency loss (see the left side of Figure 2 (b)). A pre-trained face normal estimation network N is employed to predict the face normal map of the generated image I_{out}. Intuitively, if the generated image well fits the conditioning

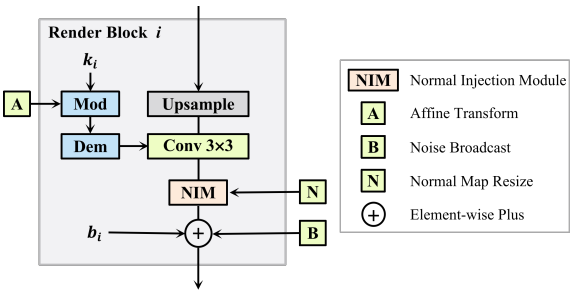


Figure 3. Details of the Render Block. The Render Block is based on the StyelGan v2 block [21], in which we introduce a Normal Injection Module to take in the resized normal map.

normal map, the face estimation network would estimate very similar face normal map to the input one. Thus we define the normal consistency loss as

$$\mathcal{L}_n = \|P(I_{\text{out}}) \odot (n - N(I_{\text{out}}))\|_1, \tag{6}$$

where I_{out} is the rendered face image, $P(\cdot)$ is a face parsing network [2] that outputs the facial region mask and makes the loss only effective on the facial region, $\odot$ denotes element-wise multiplication, and $N(\cdot)$ denotes the pre-trained face normal estimation network. We leverage SFSNet [35], which is trained with synthetic images and unlabeled real images, as the normal estimation network here.

Since the proposed neural renderer targets the face geometry reconstruction, we would like the face shape to be neatly controlled by the input face normal map but not the latent code z. Therefore, we introduce two contrastive losses to facilitate the disentanglement and to strengthen the controllability of the input normal map (see the right side of Figure 2 (b)).

On the one hand, we would like the face structure to be fully controlled by the input normal map. We construct pairs training data, in which the paired data have identical normal maps $n_1 = n_2$ but their latent codes z_1 and z_2 are different. In addition, we further adopt a facial landmark detector as a measurement of the structure consistency. Facial landmarks are essential complements to the normal maps, since normal maps focus on general structures of the surface, while the landmarks pay more attention to the facial edges and boundaries. Specifically, the facial landmark consistency loss $\mathcal{L}_{ldmk}$ is formulated as

$$\begin{aligned}&\mathcal{L}_{\text{ldmk}}(n, z_1, z_2) = \\ &\|L(G(n, z_1, \varepsilon)) - L(G(n, z_2, \varepsilon))\|_2,\end{aligned} \tag{7}$$

where L is the pre-trained landmark detector, z_1, z_2 are different latent codes fed into the renderer G.

On the other hand, we require to retain the identity of the same person when his/her pose θ and expression β vary, while the latent code and the normal map remains the same. we use a face recognition network's output features to measure whether the two output images correspond to a same person, or known as the identity loss $\mathcal{L}_{\text{id}}$,

$$\begin{aligned}&\mathcal{L}_{\text{id}}(n(\alpha, \beta_1, \theta_1), n(\alpha, \beta_2, \theta_2), z) = \\ &\|R(G(n(\alpha, \beta_1, \theta_1), z, \varepsilon)) - R(G(n(\alpha, \beta_2, \theta_2), z, \varepsilon))\|_2,\end{aligned} \tag{8}$$

where R is the pre-trained and fixed face recognition network, θ_1 and θ_2 denote the different poses fed into the renderer. The human shape α and the latent z remain the same so that we can obtain a same person with the similar facial textures from different view points.

The facial landmark consistency loss $\mathcal{L}_{\text{ldmk}}$, together with the identity loss $\mathcal{L}_{\text{id}}$, form the contrastive loss, which is designed to further disentangle the input normal map and the latent code z.

Furthermore, to enhance the quality of the rendered image, we also introduced the adversarial loss $\mathcal{L}_{\text{adv}}$. The subsequent loss function $\mathcal{L}_{\text{GAR}}$ for the training of the GAR is a weighted sum of the aforementioned losses

$$\mathcal{L}_{\text{GAR}} = \lambda_{\text{n}}\mathcal{L}_{\text{n}} + \lambda_{\text{ldmk}}\mathcal{L}_{\text{ldmk}} + \lambda_{\text{id}}\mathcal{L}_{\text{id}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} \tag{9}$$

where the λs are weights of the corresponding losses.

3.2. Face Geometry Reconstruction with Generative Adversarial Renderer

In this section, we introduce our optimization-based framework for face geometry reconstruction, aided by the proposed neural renderer.

Optimization-based face geometry reconstruction. After the neural renderer is trained, it can replace the differentiable renderer in the optimization-based face geometry reconstruction pipeline (indicated by red lines in Fig. 2(a)). Given a test face image I_{t} in the wild, our goal is to reconstruct the face geometry via optimizing the 3DMM parameters α, β and θ, which can be used to generate the normal map. The input latent code z as well as the noise ε that encodes other factors of the face image is also optimized.

We first initialize these parameters, and render them with the trained and fixed neural renderer to obtain the rendered image, which is then used to calculate the loss with I_{t}. The face geometry reconstruction loss $\mathcal{L}_f$ is defined as

$$\begin{aligned}\underset{\alpha,\beta,\theta,z}{\text{minimize}}\ &\mathcal{L}_f(\alpha, \beta, \theta, z, \varepsilon) = \|G(\tilde{n}(\alpha, \beta, \theta), z, \varepsilon) - I_{\text{t}}\|_2^2 \\ &+ \sum_i \|F_i(G(\tilde{n}(\alpha, \beta, \theta), z)) - F_i(I_{\text{t}})\|_2^2 \\ &+ \lambda_n ||\varepsilon||_2^2,\end{aligned} \tag{10}$$

where G represents the fixed generative adversarial renderer, $\tilde{n}$ is the normal map calculated from geometry coefficients (α, β, θ), F_i is the ith layer feature map by an

Method	Condition	CelebA	FFHQ
Progressive GAN [19]	×	7.79	8.04
StyleGAN [20]	×	5.17	4.40
Ours	✓	5.48	5.09

Table 1. FID scores of state-of-the-art face generation methods. Even though our GAR is trained with more conditioning constraints and the controllability of the network is promoted, our output images are still competitive to those of the unconditional StyleGAN in terms of image quality and diversity.

ImageNet-pretrained VGG network to model the perceptual loss, and λ_n weights the regularization term on the random noise. By minimizing the above face geometry reconstruction loss $\mathcal{L}_f$, we can obtain optimized geometry parameters α, β and θ.

Initialization with Renderer Inverting. Although the optimization with random initialization can produce plausible face geometry, we noticed that the gradient-based optimization is likely to get stuck at the local minima of the cost function. Inspired by [4], we design a renderer inverting network V to predict a good initial point for the gradient-based optimization of the latent code z to tackle this problem (indicated by green lines in the Fig. 2(a)).

The renderer inverting network V and the generative adversarial renderer G are trained in a coupled way, where the output of the neural renderer (the generated face image I_{out}) is input into V to convert the image back to a latent code $\hat{z}$. Ideally, the reconstructed latent code $\hat{z}$ should be close to the input latent code z.

We design the structure of the inverting network V symmetric to the GAR G, which is more theoretically interpretable, with Conv Layers converted to Deconv Layers, and the statistical mean and variance of feature maps are used to estimate the latent code z with an MLP that has same depth and channels for each layers as the style transfer MLP. The resulting feature maps of the inverting network should be of the same spatial size as those in the corresponding layer of the neural renderer. The reconstructed latent code $\hat{z}$ is estimated based on the concatenation of each layers' statistic means and standard variances followed by an MLP. The loss function for training the renderer inverting network is therefore

$$\begin{aligned}\mathcal{L}_z(R) =& \|\text{MLP}([\mu(R_i(I_{\text{out}})); \sigma(R_i(I_{\text{out}}))]) - z\|_2^2 \\ &+ \sum_i \|G_i(n, z, \theta) - R_i(I_{\text{out}})\|, \end{aligned} \tag{11}$$

where $I_{\text{out}} = G(n, z, \text{noise})$ is the generated face image from the neural renderer, R_i and G_i denotes the feature map from the ith layer of R and G respectively, μ, σ is the mean and variance of the feature map in R.

Initialization with 3DMM Solving. To obtain good initial 3DMM face parameters, we adopt the traditional 3DMM fitting algorithm [6] based on 2D facial landmarks. The loss function is accordingly defined as

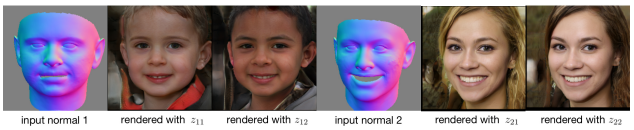


Figure 4. Generated Images with same normal and different latent codes. Our GAR renders the input normal map with diverse facial textures encoded by different latent codes.

$$\mathcal{L}_{3d} = \sum_i (kRv_i' + t - l_i) + \left|\frac{s}{\sigma_s}\right| + \left|\frac{e}{\sigma_e}\right|, \tag{12}$$

where $v_i' = v_i + W_i^s s + W_i^e e$ is the location of the vertex. α, β are the 3DMM parameters of shape and expressions, W_i^s, W_i^e represent the linear bases of shape and expression in 3DMM model, k, R, t are the pose parameters, and v_i is the i-th vertex mean position.

The renderer inverting network, together with the 3DMM parameter pre-solving, provides a good initialization for the optimization. The optimization of the 3D face shape can then be performed by minimizing the photometric loss between the rendered image and the input image.

Face editing with the Neural Renderer. The conventional face reconstruction method can be used for face image editing via modifying the recovered 3D parameters and rendering the modified face geometry. However, the editing's rendered images are generally not realistic, since the reconstruction is not accurate and the rendered images are from the conventional graphics-based renderer.

The proposed face geometry reconstruction method, together with the Generative Adversarial Renderer, provides an effective approach for face editing. Specifically, given an source image I_s, the corresponding 3DMM geometry parameters $(\alpha_s, \beta_s, \theta_s)$ can be recovered by our optimization-based framework, as well as the latent code z_s and ε_s. All or a portion of the 3DMM parameters can be chosen for editing. By rendering the edited parameters with the Generative Adversarial Renderer, we could obtain the corresponding edited face image I_t with realistic details. Even though the general idea is similar, the edited faces are much more appealing, owing to the proposed novel renderer.

4. Experiments

4.1. Dataset

Our algorithm is trained in a self-supervised manner, and requires no image annotated with 3DMM parameters. To train and test our proposed algorithm's performance, the following datasets are adopted.

Flickr-Faces-High-Quality (FFHQ) [20] is a dataset of aligned faces in resolution of 1024×1024 with labeled facial landmarks. The dataset covers larger variations of face orientations, backgrounds than other high-resolution

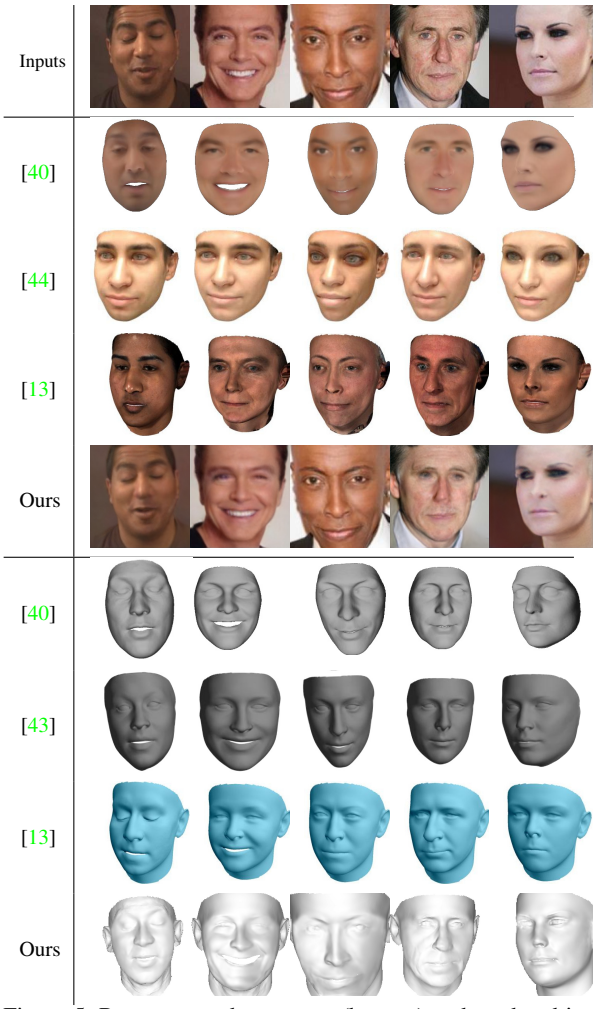


Figure 5. Reconstructed geometry (bottom) and rendered images (top) compared to previous methods. Our results outperforms others by a large margin, in both the geometry accuracy and the similarity of the rendered images.

datasets. We adopt this dataset to train our generative adversarial renderer.

CelebFaces Attribute (CelebA) [27] is a dataset of celebrities with more variations, which contains challenging cases for reconstruction. This dataset is used for the self-supervised training of the normal estimation network [35], and a subset of this dataset is used to evaluate the quality of the rendering results.

MoFA Dataset (MoFA) [41] is a combination of four datasets [9, 17, 26, 36], including unlimited faces in widely-ranged circumstances. We follow [41] and [13] to evaluate our method qualitatively on its testset.

Florence 3D Faces (Florence) [3] includes several scanned 3D face meshes of 53 subjects. The three videos about these subjects are provided, which are taken from outdoor, indoor, and cooperative environments.

Method	Cooperative		Indoor		Outdoor	
	Mean	Std	Mean	Std	Mean	Std
Tran et al.[44]	1.93	0.27	2.02	0.25	1.86	0.23
Booth et al. [7]	1.82	0.29	1.85	0.22	1.63	0.16
Genova et al. [14]	1.50	0.13	1.50	0.11	1.48	0.11
GANFIT [13]	0.95	**0.11**	0.94	0.11	0.94	0.11
Ours w/o norm-cycle	1.20	0.31	1.10	0.33	1.40	0.53
Ours w/o initial	3.20	2.10	3.21	1.97	2.98	1.43
Ours	**0.94**	0.12	**0.92**	**0.11**	**0.90**	**0.08**

Table 2. Reconstruction errors of meshes in terms of point-to-plane distance on Florence dataset.

4.2. Implementation Details.

For all of our experiments, a given face image is aligned to our fixed template using 68 landmark locations [47, 34, 39] detected by an hourglass 2D landmark detection [31]. For the normal map estimation, we adopt SFSNet [35].

During the GAR training process, we optimize parameters using Adam solver with a 0.01 learning rate. We set our balancing factors as $\lambda_{\mathrm{n}} = 2.0$, $\lambda_{\mathrm{ldmk}} = 2.0$, $\lambda_{\mathrm{id}} = 1.0$, $\lambda_{\mathrm{adv}} = 1.0$.

For the evaluation of the Florence dataset, we uniformly sample 5 frames of each video, and calculate the average of the vertex coordinates for evaluation following [13]. The evaluation metric for face reconstruction is the point-to-plane error of each vertex of reconstructed 3DMM meshes to the ground-truth scanned meshes.

4.3. Evaluation on Face Image Generation

The examples of generated images by our approach can be seen in Figure 4, which shows that the proposed Generative Adversarial Renderer can generate face images with much higher visual quality than conventional graphics-based renderers, where random hairstyle, glasses, and other attributes are well generated. To quantitatively analyze the image quality of the generated images, we randomly generated $50,000$ images with Progressive GAN [19], StyleGAN [20] and our GAR, and calculate the Frechet Inception Distance (FID) [38] between the generated images and the real image datasets. The results are presented in Table 1. It demonstrates that even though our GAR is trained with more conditioning constraints and the controllability of the network is promoted, our output images are still competitive to those of the unconditional StyleGAN in terms of image quality and diversity.

For the input conditioning normal maps in Fig. 4, we can see that the face geometry is well maintained during the rendering. This indicates that our GAR follows the important role of a renderer, faithfully converting an input normal map and a latent code to a corresponding face image.

4.4. Evaluation on Face Geometry Reconstruction

Qualitative Comparison. We qualitatively compare our reconstruction algorithm with several state-of-the-art methods

Figure 6. Face Editing Effects. By editing the 3DMM parameters, the rendered image would present corresponding attributes. In Row 1 and 2, the pose is set to turn from left to right, so the faces in the rendered images gradually changes while the identity and the expression maintain unchanged. In Row3 and 4, we present the resulting images of editing face expressions.

on the MoFA-Test dataset (see Figure 5). Rows 6 to 9 are reconstructed face meshes. Our mesh results are apparently more accurate in terms of both shape and expression, with more high-fidelity details. Row 2 to Row 5 are rendered images. Our rendered images are very close to the input images, since we significantly narrow the gap between the rendered and the realistic images.

User Study. We also conducted a user study to ask people to vote for a reconstruction result most similar to the input image. The results show that 59.1% users believe our result is the most consistent with the target image, while the second best [40] has only 28.2%, which verifies the superiority of the expression ability of our algorithm.

Quantitative Comparison. To quantitatively evaluate our reconstruction algorithm's performance, we use scanned human faces to test the accuracy. We use 5 frames from each video in the Florence Dataset [3] and compare the result to the ground-truth scanned meshes. The results are shown in Table 2. Since our mesh is calculated in the camera space, we perform an ICP (iterative-closest-point) algorithm to align the output mesh by our method to the scanned ground-truth. The errors are calculated as the point-to-plane distance for each vertex on our reconstructed meshes. Our method has a better result in terms of the average error.

4.5. Ablation Study

In this section, we present the results of an ablation study on investigating different components of our proposed face reconstruction framework.

Effect of Normal Consistency Loss. As shown in Line 5 and Line 7 of the Table 2, when training GAR without the Normal Consistency Loss ("Ours w/o norm-cycle"), the training cannot guarantee the results of the generator to well condition on the input normal maps. This indicates that the proposed Normal Consistency Loss is significant in promoting the controllability of the input normal maps.

Effect of Renderer Inverting Initialization. As shown in Line 6 and Line 7 of the Table 2, when the latent code is not initialized with the renderer inverting ("Ours w/o initial"), the reconstruction error shows a severe increase and might not converge for specific faces.

Choices of the Conditions. Previous works tend to employ 3DMM parameters or depth as the conditioning inputs. However, we found that the face normal map is a more effective form of condition for our GAR. With the face normal maps as inputs, the loss is minimized more stably and faster, and the network could converge to a better optimum.

Effect of the Normal Injection Module (NIM). Comparing with the simple concatenation of the normal map into the feature maps, the proposed NIM is effective in further minimizing the loss value, which demonstrates the effectiveness of the proposed NIM.

4.6. Qualitative Evaluation on Face Image Editing

As mentioned in Section 3.2, our method is capable of face editing. As shown in Figure 6, by editing the 3DMM parameters, the rendered image would present corresponding attributes. In the first row, the pose is set to turn from left to right, so the faces in the rendered images gradually change while the identity and the expression maintain unchanged. In the second and third rows, we present the resulting images of editing facial expressions. For results please refer to the supplementary materials.

5. Conclusion and Future Work

In this paper, we propose a Generative Adversarial Renderer (GAR) that takes a normal map and a latent code and outputs a rendered face image. Based on GAR, we also propose an optimization-based face geometry reconstruction method, as well as an initialization method by Renderer Inverting.

The main idea of this paper may be naturally extended to arbitrary scenarios. For instance, we may train a generative adversarial renderer for bedrooms, which takes normal maps of a bed and then renders corresponding images. The renderer might also be used to reconstruct their geometry.

Acknowledgement This work is supported in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants (Nos. 14208417 and 14207319), in part by CUHK Strategic Fund.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4432–4441, 2019. 4

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 5

[3] Andrew D Bagdanov, Alberto Del Bimbo, and Iacopo Masi. The florence 2d/3d hybrid face dataset. In *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding*, pages 79–80. ACM, 2011. 7, 8

[4] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4502–4511, 2019. 2, 4, 6

[5] Volker Blanz, Curzio Basso, Tomaso Poggio, and Thomas Vetter. Reanimating faces in images and video. In *Computer graphics forum*, volume 22, pages 641–650. Wiley Online Library, 2003. 1, 3

[6] Volker Blanz and Thomas Vetter. 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. ACM Press/Addison-Wesley Publishing Co., 1999. 3, 6

[7] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3d face morphable models" in-the-wild". In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5464–5473. IEEE, 2017. 7

[8] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):43, 2014. 1, 3

[9] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014. 3, 7

[10] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *IEEE Computer Vision and Pattern Recognition*, 2020. 1, 2, 3

[11] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2019. 1

[12] Baris Gecer, Alexandros Lattas, Stylianos Ploumpis, Jiankang Deng, Athanasios Papaioannou, Stylianos Moschoglou, and Stefanos Zafeiriou. Synthesizing coupled 3d face modalities by trunk-branch generative adversarial networks. In *European Conference on Computer Vision*, pages 415–433. Springer, 2020. 3

[13] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1164, 2019. 1, 2, 3, 7

[14] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018. 1, 3, 7

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[16] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (TOG)*, 34(4):125, 2015. 3

[17] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008. 7

[18] Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. Disentangled representation learning for 3d face shape. *arXiv preprint arXiv:1902.09887*, 2019. 3

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 6, 7

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 3, 4, 6, 7

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 3, 4, 5

[22] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018. 1, 2, 3

[23] Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou. Avatarme: Realistically renderable 3d facial reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 760–769, 2020. 3

[24] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (TOG)*, 36(6):194, 2017. 3

[25] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. *arXiv preprint arXiv:1901.05567*, 2019. 2

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 7

[27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018. 7

[28] Nadia Magnenat-Thalmann, E Primeau, and Daniel Thalmann. Abstract muscle action procedures for human face animation. *The Visual Computer*, 3(5):290–297, 1988. 3

[29] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1, 3

[30] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, Hao Li, Richard Roberts, et al. pagan: real-time avatars using dynamic textures. *ACM Trans. Graph.*, 37(6):258–1, 2018. 3

[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 7

[32] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. *arXiv preprint arXiv:1904.01326*, 2019. 4

[33] Stylianos Ploumpis, Evangelos Ververas, Eimear O'Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick Pears, William Smith, Baris Gecer, and Stefanos P Zafeiriou. Towards a complete 3d morphable model of the human head. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3

[34] Shengju Qian, Keqiang Sun, Wayne Wu, Chen Qian, and Jiaya Jia. Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10153–10163, 2019. 7

[35] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. 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, 2018. 5, 7

[36] Jie Shen, Stefanos Zafeiriou, Grigoris G Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 50–58, 2015. 7

[37] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. *arXiv preprint arXiv:1907.10786*, 2019. 4

[38] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, 2018. 7

[39] Keqiang Sun, Wayne Wu, Tinghao Liu, Shuo Yang, Quan Wang, Qiang Zhou, Zuochang Ye, and Chen Qian. Fab: A robust facial landmark detection framework for motion-blurred videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5462–5471, 2019. 7

[40] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2549–2559, 2018. 7, 8

[41] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. 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, 2017. 7

[42] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphable model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1126–1135, 2019. 3

[43] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7346–7355, 2018. 3, 7

[44] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. 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, 2017. 7

[45] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *ECCV*, 2020. 1

[46] Qianyi Wu, Juyong Zhang, Yu-Kun Lai, Jianmin Zheng, and Jianfei Cai. Alive caricature from 2d to 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7336–7345, 2018. 3

[47] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2138, 2018. 7

[48] Hao Zhu, Chaoyou Fu, Qianyi Wu, Wayne Wu, Chen Qian, and Ran He. Aot: Appearance optimal transport based identity swapping for forgery detection. In *Neural Information Processing Systems (NeurIPS)*, 2020. 1

[49] Wenbin Zhu, HsiangTao Wu, Zeyu Chen, Noranart Vesdapunt, and Baoyuan Wang. Reda: Reinforced differentiable attribute for 3d face reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4958–4967, 2020. 1, 3

[50] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155, 2016. 1, 3