# Towards High-Fidelity Face Self-Occlusion Recovery via Multi-View Residual-Based GAN Inversion

**Jinsong Chen,**[1,2] **Hu Han,**[1,2,3,*] **Shiguang Shan**[1,2]

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Pengcheng National Laboratory, Shenzhen 518055, China
chenjinsong20@mails.ucas.ac.cn, {hanhu, sgshan}@ict.ac.cn

## Abstract

Face self-occlusions are inevitable due to the 3D nature of the human face and the loss of information in the projection process from 3D to 2D images. While recovering face self-occlusions based on 3D face reconstruction, e.g., 3D Morphable Model (3DMM) and its variants provides an effective solution, most of the existing methods show apparent limitations in expressing high-fidelity, natural, and diverse facial details. To overcome these limitations, we propose in this paper a new generative adversarial network (MvInvert) for natural face self-occlusion recovery without using paired image-texture data. We design a coarse-to-fine generator for photorealistic texture generation. A coarse texture is computed by inpainting the invisible areas in the photorealistic but incomplete texture sampled directly from the 2D image using the unrealistic but complete statistical texture from 3DMM. Then, we design a multi-view Residual-based GAN Inversion, which re-renders and refines multi-view 2D images, which are used for extracting multiple high-fidelity textures. Finally, these high-fidelity textures are fused based on their visibility maps via Poisson blending. To perform adversarial learning to assure the quality of the recovered texture, we design a discriminator consisting of two heads, i.e., one for global and local discrimination between the recovered texture and a small set of real textures in UV space, and the other for discrimination between the input image and the re-rendered 2D face images via pixel-wise, identity, and adversarial losses. Extensive experiments demonstrate that our approach outperforms the state-of-the-art methods in face self-occlusion recovery under unconstrained scenarios.

## Introduction

Face imaging is the process of projecting a 3D human face to a 2D plane. Therefore, some parts of the face can be occluded by another part of the face during the projection process. Self-occlusion seriously affects the performance of succeeding face analysis, face recognition, and face reenactment tasks.

Face self-occlusion recovery is an ill-posed problem because of insufficient constraints. Existing methods for self-occlusion recovery are generally based on 3DMM (Blanz and Vetter 1999) and its variants. The early approaches deal with face self-occlusion by using the statistically complete texture represented by a linear combination of the basis texture vectors in 3DMM (Yin et al. 2006; Cao et al. 2013; Koppen et al. 2018; Booth et al. 2016). While such a texture is complete, it is not photorealistic compared with the input face image; particularly it cannot express natural and diverse face appearance details.

To resolve the unrealistic face texture issue, many latter approaches propose to sample a texture from the input 2D face image (UV texture map) to replace the linearly combined texture (PCA-texture) in 3DMM. However, because of self-occlusion, the UV sampled face texture is incomplete. To address this issue, some studies aim to generate a complete and photorealistic face texture based on the incomplete one. The most common method is to learn a mapping between incomplete UV sampled textures to real complete textures, which usually relies on paired data such as 2D face image and its corresponding complete face texture (Deng et al. 2018). The complete face textures may come from 3D scanning or computed from multi-view 2D face images via photometric stereo. Apparently, such kinds of paired data are still limited because of the high cost of data acquisition.

To overcome the above limitations, recent approaches propose to leverage generative models, such as a GAN, to model the distribution of the complete and high-fidelity feature textures (Gecer et al. 2019; Lee et al. 2020b). Specifically, a GAN can be trained using a complete high-fidelity texture dataset so that it can generate arbitrary high-fidelity textures by changing the latent vector. Then, instead of optimizing the linear texture combination coefficients in early 3DMM based approaches, a complete high-fidelity face texture can be generated to replicate the input 2D face image by optimizing the latent vector. Another method to get rid of the constraint of requiring paired data is to leverage the prior knowledge in pre-trained models on large 2D face datasets to perform inpainting in 2D face image space. For example, (Gecer, Deng, and Zafeiriou 2021) proposed to render multi-view 2D face images based on incomplete high-fidelity textures, and used StyleGAN (Karras, Laine, and Aila 2019) to perform inpainting for the rendered 2D image so that the corrupted part in these images can be recovered. Then, these recovered 2D images can be merged, e.g., via alpha blending, to obtain a complete high-fidelity face texture. While these face self-occlusion recovery methods can obtain high-


*Corresponding author

Figure 1: Input portaits with self-occlusion and corresponding high-fidelity characters generated by our face self-occlusion recovery method. Our method is robust to lighting and pose, and can faithfully restore personalized details like skin tone, and wrinkles.

quality face texture, their computational costs are high compared with the methods that learn a mapping between incomplete UV texture to complete texture.

In this paper, we propose an efficient GAN-based face self-occlusion recovery method which does not require paired image-texture data for training. We first use a CNN-based 3DMM model to obtain the 3D face shape and the statistically complete face texture via a linear combination of texture basis (PCA-texture). At the same time, we sample a UV face texture from the input 2D image, which is photorealistic but corrupted in the self-occlusion areas. Then, we compute a coarse face texture by performing inpainting for the incomplete high-fidelity texture by using the corresponding areas from the PCA-texture via Poisson blending. Next, our coarse-to-fine texture generation leverages multi-view rendering and residual-based GAN inversion, followed by Poisson blending to obtain the final complete and high-fidelity texture (see Fig. 1). We design our discriminator with two heads, e.g., one for discrimination between rendered and real face images, and the other between re-rendered 2D face images and real 2D face images.

The contributions of this work are as follows:

- While our approach falls into the second category of methods, i.e., aiming at obtaining a complete high-fidelity texture in UV space, our method differs from existing methods in that it does not require paired image-texture data to perform fully supervised training, thus making it possible to leverage face images in the wild to perform face self-occlusion recovery.
- We propose a coarse-to-fine texture refinement approach, which performs multi-view GAN inversion to obtain high-fidelity 2D images and their textures, followed by Poisson blending to obtain a complete and high-fidelity face texture.
- We design our discriminator considering both texture-level and image-level adversarial learning.
- We also design a differentiable screened Poisson blending equation solving method so that the whole network can be trained end-to-end with significantly reduced computational cost.

## Related Work

### 3D Face Reconstruction

Face self-occlusion recovery is related to 3D face reconstruction and image restoration. We briefly review the related methods below. As we summarized in the introduction, through 3D face reconstruction, a complete face texture can usually be obtained. For example, the early 3D face reconstruction method 3D Morphable Model (3DMM) and its variants usually use a PCA-based 3D shape model and a texture model, as well as a spherical harmonics illumination model to represent a 3D face. The early 3DMM methods solve the shape, texture, illumination parameters via solving a nonlinear optimization problem (Richardson et al. 2017; Booth et al. 2017). This process is often slow and computationally expensive. With the development of Convolutional Neural Networks, recent studies (Richardson et al. 2017; Guo et al. 2018; Deng et al. 2019b; Shang et al. 2020) utilize DCNN to predict 3DMM parameters.

### Face Image Recovery

Face image recovery involves face image de-occlusion, inpainting, denoising, super-resolution among others. Here, we only briefly review the face de-occlusion and inpainting methods related to face self-occlusion recovery. Because of the great success of GAN in image generation and translation, GAN has also been used for face texture recovery, de-occlusion, inpainting, etc (Deng et al. 2018; Gecer et al. 2019; Tran and Liu 2018; Lee and Lee 2020). Deng et

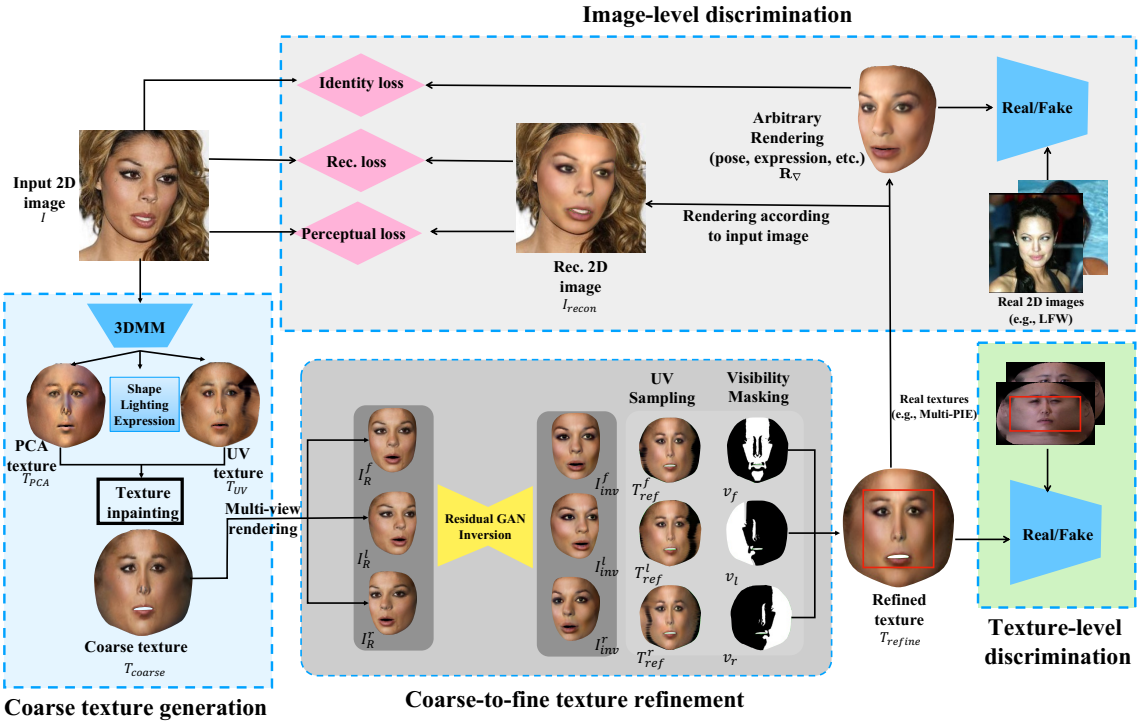


Figure 2: The overall architecture of the proposed approach (MvInvert) for high-fidelity face self-occlusion recovery via multi-view residual-based GAN inversion.

al. (Deng et al. 2018) obtain complete UV texture by fitting a 3DMM to various multi-view images and train a GAN to complete the self-occluded parts in UV texture. Gecer et al. (Gecer et al. 2019) build a generative UV texture model based on GAN, and use it to replace the PCA-based texture model in 3DMM. Although these methods can produce high-fidelity textures, they require a large-scale dataset with high-quality UV textures, which is often not publically available. Instead of using real UV texture, Tran et al. (Tran and Liu 2018) utilize synthetic 3D face images to perform 3D face recognition, where self-occluded textures are recovered via interpolation.

Yin et al. (Yin et al. 2021) and Lin et al. (Lin, Yuan, and Zou 2021) bypass the dependency on real UV textures by using pseudo UV textures for training. They use CNN-based 3DMM to predict the shape and pose coefficients for face shape reconstruction on large-scale in-the-wild face images. Then they sample from input images according to the 3D vertexes - 2D image correspondence to obtain high-fidelity but incomplete textures. To get a complete pseudo UV maps without self-occlusions, Yin et al. (Yin et al. 2021) repair the incomplete textures using the corresponding PCA-texture by 3DMM via seamless image blending, while Lin et al. (Lin, Yuan, and Zou 2021) blend them with mean skin color computed from the input face skin and fill in the occluded regions using symmetry. Zhou et al. (Zhou et al. 2020) produce arbitrary face rotations from a single image to serve as a strong self-supervision. While these methods do not require collecting real UV texture data for training, the recovered face texture may have Gibbs artifacts because of the use of pseudo UV textures. Lin et al. (Lin et al. 2020) make radical changes by replacing CNN with Graph Convolutional Network (GCN) to refine the 3DMM texture from the features extracted with a face recognition model. Though their method is trained solely on in-the-wild face images, it is limited by the highly abstract nature of the identity features, leading to a lack of details in the reconstructed textures.

## GAN Inversion

Recent GANs (Karras et al. 2017; Brock, Donahue, and Simonyan 2018; Karras, Laine, and Aila 2019; Zhu et al. 2017a,b) are able to generate considerably high-resolution and more photorealistic images than the conventional DCGAN architecture (Yu et al. 2017). The high-quality image generalization ability of their generators can be exploited by learning a mapping from the image to the latent space of the encoder (a.k.a. GAN inversion). These approaches can be classified into three groups: optimization-based methods (Zhu et al. 2016; Ma, Ayaz, and Karaman 2019; Abdal, Qin, and Wonka 2019; Shen et al. 2020; Abdal, Qin, and Wonka 2020; Zhu et al. 2020), encoder-based methods (Richardson et al. 2021; Tov et al. 2021), and residual encoder-based methods (Alaluf, Patashnik, and Cohen-Or 2021). Optimization-based methods project the input face image to the latent space $\mathcal{W}$ by directly optimizing the latent code until the reconstruction error between the input face image and the generation produced by a generator is minimized. Encoder-based methods learn an encoder to pre-

dict the latent code from the face image. Residual encoder-based methods learn the inverse mapping by minimizing the disparity between the input image and the generation by iteratively refining the residual codes predicted by an encoder. Although optimization-based methods can produce the most realistic face UV map, they are usually complicated and have high computational costs. By contrast, the residual encoder-based methods have much higher efficiency. For example, while OSTec (Gecer, Deng, and Zafeiriou 2021) takes 5 minute to generate a UV texture from a 2D face, GANFit (Gecer et al. 2019) takes only 30 seconds.

## Approach

Fig. 2 shows an overview of the proposed method, which consists of three essential components: coarse texture generation, coarse-to-fine texture refinement, texture- and image-level discrimination. We detail each component below.

### Coarse Texture Generation via 3DMM

Given an input 2D face image $\mathbf{I}$, we first recover the 3D face elements such as shape ($\mathbf{S}_{\text{3D}}$), texture ($\mathbf{T}_{\text{PCA}}$), pose, etc, by using a CNN-based 3DMM (Shang et al. 2020). CNN-based 3DMM uses a CNN to predict the linear combination coefficients, and uses the predicted coefficients to compute the 3D shape and texture as follows:

$$\begin{aligned} \mathbf{S}_{\text{3D}} &= \bar{\mathbf{S}} + \mathbf{C}_s \mathbf{S}_{Base} + \mathbf{C}_e \mathbf{E}_{Base} \\ \mathbf{T}_{\text{PCA}} &= \bar{\mathbf{T}} + \mathbf{C}_T \mathbf{T}_{Base} \end{aligned} \tag{1}$$

where $\mathbf{C}_s$, $\mathbf{C}_e$ and $\mathbf{C}_T$ denote the predicted linear combination coefficients for 3D shape, expression and texture respectively. $\bar{\mathbf{S}}$ and $\bar{\mathbf{T}}$ denote the average 3D shape and texture respectively; $\mathbf{S}_{Base}$ and $\mathbf{T}_{Base}$ are the PCA bases of the shape and texture, respectively.

Given such a 3D reconstruction result, we can also compute a UV sampled face texture ($\mathbf{T}_{\text{UV}}$) based on the correspondence between 3D face vertexes and 2D facial landmarks. As we reviewed in the introduction, the UV sampled face texture $\mathbf{T}_{\text{UV}}$ is photorealistic but incomplete in the self-occlusion areas. By contrast, the PCA face texture $\mathbf{T}_{\text{PCA}}$ is complete but not photorealistic. Therefore, we propose to build a coarse texture $\mathbf{T}_{\text{coarse}}$ by performing inpaining for $\mathbf{T}_{\text{UV}}$ by using the corresponding areas in $\mathbf{T}_{\text{PCA}}$ via Poisson blending (Arnaud Dessein and Richard C Wilson 2014).

### Coarse-to-fine Texture Refinement via Multi-view GAN Inversion

**Multi-view GAN Inversion** We propose a *multi-view GAN inversion (MvInvert)* to perform coarse-to-fine texture refinement for $T_{\text{coarse}}$. Specifically, we first perform rendering using coarse texture under three different poses such as frontal, left, and right, and obtain rendered 2D images ($\mathbf{I}_R^f$, $\mathbf{I}_R^l$, and $\mathbf{I}_R^r$) as shown in Fig. 2. Since the coarse texture $T_{\text{coarse}}$ is still not perfect, these rendered 2D images also contain some artifacts such as blurriness, inconsistency, non-natural, etc. For each rendered 2D face image, we then utilize GAN inversion to exploit the prior image distribution knowledge learned by generative GAN modes to recover a high-quality image. Similar to ReStyle Encoder (Alaluf, Patashnik, and Cohen-Or 2021), we also use a residual-based latent encoder to predict the residuals instead of the inverted latent code:

$$\begin{aligned} &\Delta_i = \mathbf{E}(\mathbf{I}_R^i) \\ &\mathbf{w}_i \leftarrow \Delta_i + \mathbf{w}_0, i \in \{f, l, r\} \end{aligned} \tag{2}$$

where $\mathbf{E}(\cdot)$ is the residual-based encoder, $\Delta_i$ is the predicted latent residual corresponding to the rendered image with a novel camera view, $\mathbf{w}_0$ denotes the mean latent code computed from the large-scale face image dataset and $\mathbf{w}_i$ is the final latent parameter for each rendered image.

We then feed the learned latent parameters into the pre-trained StyleGANv2 generator to obtain the high-fidelity natural face images $\mathbf{I}_{inv}^i$ for further view synthesis.

$$\mathbf{I}_{inv}^i = \mathbf{G}(\mathbf{w}_i), i \in \{f, l, r\} \tag{3}$$

where $\mathbf{I}_{inv}^i$ has the same size as $\mathbf{I}_R^i$.

Given the high-quality images, we perform UV sampling again based on 3D vertexes and 2D image correspondence, and obtain three texture images $\mathbf{T}_{\text{ref}}^f$, $\mathbf{T}_{\text{ref}}^l$, and $\mathbf{T}_{\text{ref}}^r$. Since $\mathbf{I}_{\text{inv}}^i$ is expected to be perfect after GAN inversion, $\mathbf{T}_{\text{ref}}^i$ can also be expected to have improved quality than $\mathbf{T}_{\text{coarse}}$.

**Masking and Stitching** The refined multi-view $\mathbf{T}_{\text{ref}}^f$, $\mathbf{T}_{\text{ref}}^l$, and $\mathbf{T}_{\text{ref}}^r$ are complementary with other. Then, we can integrate these refined textures to obtain a more complete face texture with high quality.

In order to stitch the three textures under different novel views, we first compute a visibility score map for each texture according to the estimated face shape and its pose. This visibility score of each vertex can be defined in terms of the angle between its normal and its coordinates $\mathbf{S}_i$ relative to the camera $\mathbf{c}$. We take the dot product between the unitized vertex normals and view vectors as the visibility score

$$\mathbf{V}_i = \left( \frac{[\mathbf{S}_i - \mathbf{c}_i]}{\|[\mathbf{S}_i - \mathbf{c}_i]\|_2} \cdot \mathcal{N}\left(\mathbf{S_i}\right)^T \right), i \in \{f, l, r\} \tag{4}$$

where $\mathcal{N}$ denotes the normals of the vertices. If the vertex is not visible in one view, i.e., the corresponding visibility score is smaller than 0, the score is set to zero. We then set a threshold distance of the occlusion boundary following AlbedoMM (Smith et al. 2020). If the vertex is projected within the occlusion boundary's threshold distance, we set its visibility score to zero to avoid sampling background onto the mesh. We define the per-triangle confidence value as the minimum per-vertex visibility score for all three vertexes in the triangle. We define a selection matrix for each view which selects a triangle if the view $c_i$ has the highest weight for that triangle:

$$\begin{aligned} &\left(\tilde{\mathbf{V}}_{c_i}^{tri}\right)_j = 1, \text{ if } w_j^{c_i} > w_j^u, \forall u \in \mathcal{C} \backslash \{c_i\}, \\ &i \in \{f, l, r\}, i \in \{1 \dots t\} \end{aligned} \tag{5}$$

We also construct an additional selection matrix $\tilde{\mathbf{V}}_{c_{k+1}}^{tri}$ to pick all triangles that are not chosen in any view to ensure that each triangle is selected precisely once. We similarly

define per-vertex selection matrices $\tilde{\mathbf{V}}_{c_i}^{ver} \in \{0,1\}^{m_c \times n}$ that select the vertices for which view v has the highest per-vertex weights. Thus, the screened Poisson equation for the unknown per-vertex albedo maps $\mathbf{T}_{\text{refine}} \in \mathbb{R}^{n \times 3}$ can be written as follow:

$$\begin{bmatrix} \left(\mathbf{I}_3 \otimes \tilde{\mathbf{V}}_{c_1}\right)\mathbf{G} \\ \vdots \\ \left(\mathbf{I}_3 \otimes \tilde{\mathbf{V}}_{c_k}\right)\mathbf{G} \\ \lambda\tilde{\mathbf{V}}_{c_1} \end{bmatrix} \mathbf{T}_{\mathbf{refine}} = \begin{bmatrix} \left(\mathbf{I}_3 \otimes \tilde{\mathbf{V}}_{c_1}\right)\mathbf{G}\mathbf{I}^{c_1} \\ \vdots \\ \left(\mathbf{I}_3 \otimes \tilde{\mathbf{V}}_{c_k}\right)\mathbf{G}\mathbf{I}^{c_k} \\ \mathbf{0}_{3t \times 3} \\ \lambda\tilde{\mathbf{V}}_{c_1}\mathbf{I}^{c_1} \end{bmatrix} \tag{6}$$

where $\otimes$ is the Kronecker product, $\mathbf{I}_3$ is the $3 \times 3$ identity matrix and $\mathbf{G} \in \mathbb{R}^{3t \times n}$ computes the per-triangle gradient in the $x$, $y$ and $z$ directions of a function defined on the $n$ vertices of the mesh. We solve (6) using the least square method so that $\mathbf{T}_{\text{refine}}$ seeks to match the selected gradients in each triangle. Triangles with no selected view are assumed to have zero gradients. Original view $v_0$ is chosen as the reference to resolve color offset indeterminacies, and $\lambda$ is the screening weight. We use $k = 3$ views and we set $\lambda = 0.1$ following (Smith et al. 2020).

## Overall Loss

Within the training process, we design the loss functions from two perspectives: image-level and texture-level.

**Reconstruction Loss** We use the pixel-level loss to encourage low-level semantic similarity in the visible part of the image, i.e., between the real 2D face images and re-rendered face images using our refined face texture $\mathbf{T}_{refine}$. Specifically, we compute a textured 3D face $\mathbf{S}_{\text{3D}}$ with the same expression, posture, and illumination as the input 2D image. We then project it to 2D space and blend it with the input face image $\mathbf{I}$ to get a reconstructed face image $\mathbf{I}_{recon}$. To this end, the $\ell_2$ loss can be computed as:

$$\mathcal{L}_{rec}(\mathbf{I}, \mathbf{I}_{recon}) = \|\mathbf{I} - \mathbf{I}_{recon}\|_2 \tag{7}$$

The reconstruction loss makes it possible to use self-supervision to improve self-occlusion recovery results.

**Perceptual Loss** Besides the pixel-level reconstruction loss, we also use a perceptual loss between the original 2D image and the reconstructed image for self-supervision to assure the other factors like skin tone remain unchanged. We minimize the normalized Euclidean distance of intermediate activation in a face recognition network (FaceNet (Schroff, Kalenichenko, and Philbin 2015) pretrained on CASIA-WebFace (Yi et al. 2014)) between two images:

$$\mathcal{L}_{perc}(\mathbf{I}, \mathbf{I}_{recon}) = \sum_{j}^{n} \frac{\|F_j(\mathbf{I}) - F_j(\mathbf{I}_{recon})\|_2}{H_{F_j} \times W_{F_j} \times C_{F_j}} \tag{8}$$

where $F_j(\cdot)$ denotes j-th layer of the deep feature encoding and the $H_{F_j}$, $W_{F_j}$ and $C_{F_j}$ the height, weight and channel number of the activation output respectively.

**Identity Loss** For identity loss, we compute the cosine similarity between the deep feature vectors, extracted with a pre-trained ArcFace (Deng et al. 2019a), of multiple arbitrarily rendered 2D face images (with random expression, pose, and lighting) from $\mathbf{T}_{\text{refine}}$ and the deep feature vector of the input 2D images:

$$\mathcal{L}_{id}(\mathbf{I}, \mathcal{R}_\nabla) = \frac{1}{K}\sum_{k=1}^{K}\left(1 - \frac{F^n(\mathbf{I}).F^n\left(\mathbf{R}_\nabla^k\right)}{\|F^n(\mathbf{I})\|_2 \left\|F^n\left(\mathbf{R}_\nabla^k\right)\right\|_2}\right) \tag{9}$$

where $\mathcal{R}_\nabla$ are the $K$ 2D images rendered with the random pose, expression, and illumination. We calculate the average identity loss between each rendered image $\mathbf{R}_\nabla^k$ and the input image $\mathbf{I}$. Identity loss ensures that our texture completion and refinement process do not change the identity.

**Adversarial Loss** To encourage more photorealistic results for our coarse-to-fine texture refinement via multi-view GAN inversion, we design a two-head discriminator, including one for global and local discrimination between the recovered UV textures (e.g., $\mathbf{T}_{\text{refine}}^i$) and a set of real UV textures (e.g., $\mathbf{T}_{\text{real}}^k = 1^K$), and the other for discrimination between a set of authentic face images in the wild (e.g., $\mathbf{I}_{\text{real}}^l = 1^L$) and the arbitrarily re-rendered face images $\mathcal{R}_\nabla$. We train the discriminators to determine whether the generated outputs are real or false, while our generators are trained to deceive the discriminators. The adversarial losses for the two heads are defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{adv}}^{\text{tex}} &= \mathbb{E}[\log D_{\text{glb}}^{\text{tex}}(\mathbf{T}_{\text{real}})] + \mathbb{E}[\log(1 - D_{\text{glb}}^{\text{tex}}(\mathbf{T}_{\text{refine}}))] \\ &+ \mathbb{E}[\log D_{\text{ctr}}^{\text{tex}}(\mathbf{T}_{\text{real}}^{\text{ctr}})] + \mathbb{E}[\log(1 - D_{\text{ctr}}^{\text{tex}}(\mathbf{T}_{\text{refine}}^{\text{ctr}}))]\end{aligned} \tag{10}$$

$$\mathcal{L}_{\text{adv}}^{\text{img}} = \mathbb{E}[\log D^{\text{face}}(\mathbf{I}_{\text{real}})] + \mathbb{E}[\log(1 - D^{\text{face}}(\mathbf{I}_{\text{refine}}))] \tag{11}$$

where $D_{\text{glb}}^{\text{tex}}$ and $D_{\text{ctr}}^{\text{tex}}$ denote global and centeral textur discriminators, and $D^{\text{face}}$ face discriminator.

# Implementation Details

Before training, all 2D face images are aligned following the method of (Bulat and Tzimiropoulos 2017). Then, we use a modified BiSeNet (Yu et al. 2018) pre-trained on CelebAMask-HQ dataset (Lee et al. 2020a) for face segmentation and facial region mask prediction. The face image datasets we used for training are CelebA (Liu et al. 2018) and FFHQ dataset collected by (Karras, Laine, and Aila 2019). We use the 3D morphable face model of Basel Face Model (Blanz and Vetter 1999) in 3D shape and coarse texture generation, and a CNN-based 3DMM regressor which is pre-trained in (Shang et al. 2020). We use the weights of pre-trained Restyle Encoder (Alaluf, Patashnik, and Cohen-Or 2021) to initialize our residual-based latent encoder. For the face generator, we use the off-the-shelf pre-trained StyleGANv2 generator (Karras et al. 2020). We set the input image size to $224 \times 224$ and the number of vertices and triangle faces to $35,709$ and $70,897$ respectively, the same as

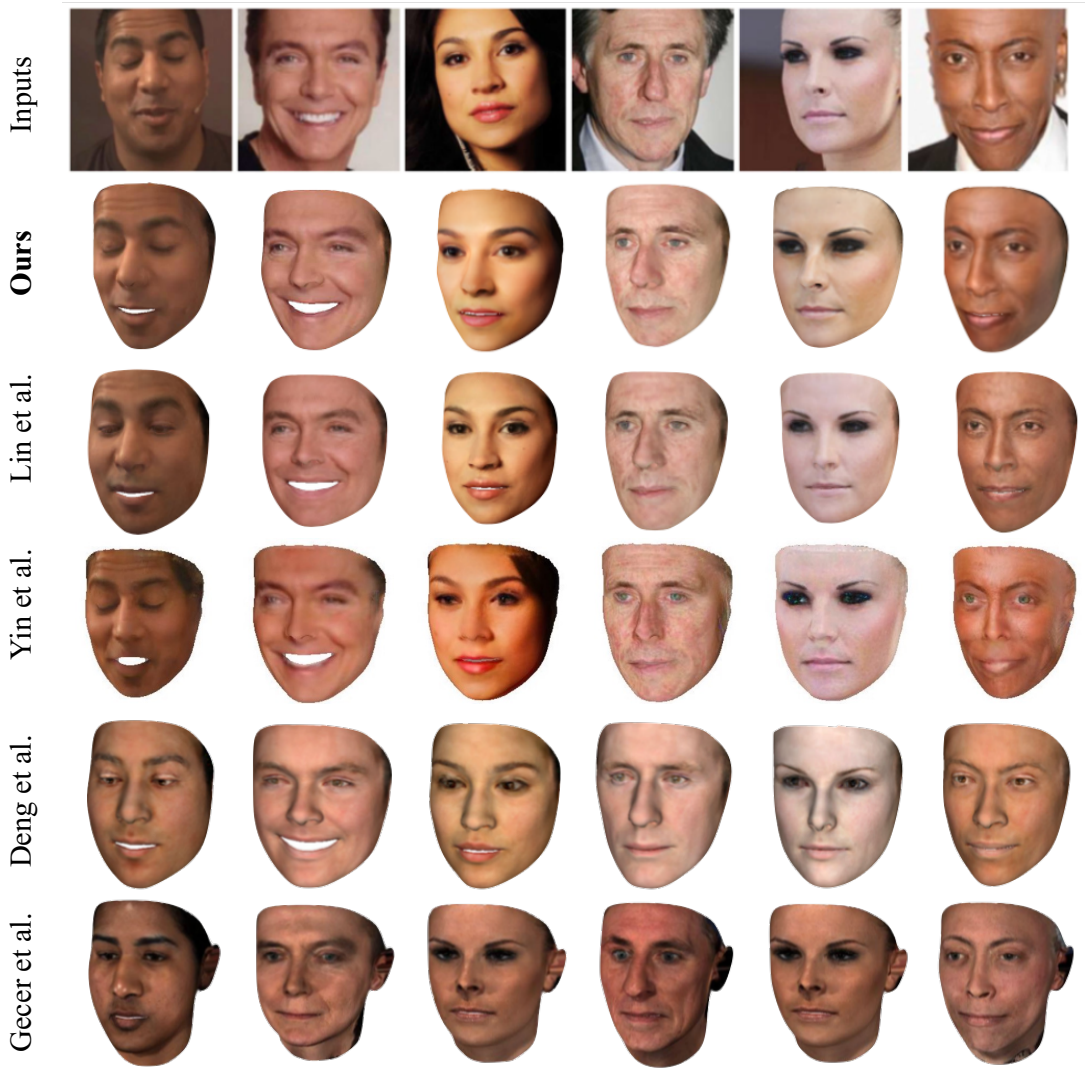


Figure 3: Qualitative result comparisons between our method and the state-of-art methods: (Lin et al. 2020), (Yin et al. 2021), (Deng et al. 2019b) and (Gecer et al. 2019) in MoFA-Test dataset (zoom in for better comparison). Please also see our dynamic results in supplementary material.

(Shang et al. 2020). The overall training loss is:

$$\begin{aligned}\mathcal{L} =& \lambda_{\text{rec}} \left(\mathcal{L}_{\text{rec}}(\mathbf{I}_0, \mathbf{I}_{\text{recon}})\right) + \lambda_{\text{perc}} \left(\mathcal{L}_{\text{perc}}(\mathbf{I}_0, \mathbf{I}_{\text{recon}})\right) \\ &+ \lambda_{\text{id}} \left(\mathcal{L}_{\text{perc}}(\mathbf{I}_0, R_{\nabla})\right) \\ &+ \lambda_{\text{adv}} \left(\mathcal{L}_{\text{adv}}^{\text{tex}} + \mathcal{L}_{\text{adv}}^{face}\right)\end{aligned} \tag{12}$$

We set hyper-parameters $\lambda_{\text{rec}} = 1.9$, $\lambda_{\text{perc}} = 0.2$ following (Deng et al. 2019b), and the other hyper-parameters empirically: $\lambda_{\text{id}} = 0.8$, $\lambda_{\text{adv}} = 0.1$. We implement our method with torch(1.7.1) and PyTorch3D (v0.4.0), and run our experiments on NVIDIA 1080Ti GPUs with Intel 2.1GHz CPUs. During inference, our network takes 3.7s to produce a refined UV texture of $35,709 \times 3$.

## Experiments Results

### Qualitative Results

We compare our method with several state-of-art methods, including 3D Face GCN (Lin et al. 2020), Deep 3D Face Reconstrution (Deng et al. 2019b), GANFit (Gecer et al. 2019) and the method in (Yin et al. 2021) on a small subset of MOFA-test dataset (shown in Fig. 3), which are widely used by existing methods. The method of (Lin et al. 2020) released the training code; so we can use the same settings to train their method. The rest of the above methods do not release their code or pre-trained models. Therefore, we can only be able to compare with the results reported in their papers. The results are shown in Fig. 3. Among the baselines, the results by (Lin et al. 2020) give the best visual quality. However, compared to state-of-the-art algorithms, our method can better retain the original input image's facial de-

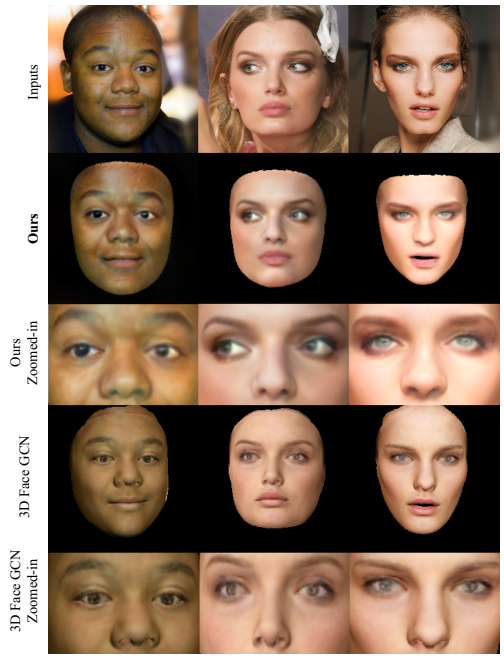


Figure 4: Zoomed-in comparison with 3D Face GCN (Lin et al. 2020) in terms of facial detail preservation.

| Method | (Deng et al. 2019b) | (Lin et al. 2020) | Ours |
|---|---|---|---|
| $L_1$ distance ↓ | 0.052 | 0.034 | **0.028** |
| PSNR ↑ | 26.58 | 29.69 | **30.78** |
| SSIM ↑ | 0.826 | 0.894 | **0.897** |
| evoLVe ↑ | 0.641 | 0.848 | **0.878** |
| LightCNN ↑ | 0.724 | 0.900 | **0.926** |

Table 1: Quantitative comparison with two SOTA methods in terms of reconstruction and identity-preserving abilities.

tails like wrinkles, local component shapes, and with fewer checkerboard artifacts. Some baseline methods like (Yin et al. 2021) may cause inconsistent texture colors and are not able to handle foreign object occlusion.

## Quantitative Results

Due to the lack of ground-truth 3D face data, we follow the widely used metrics defined based on reconstructed 2D face images and the original face images. We evaluate our approach in two folds: the reconstruction ability and the identity-preserving ability.

With regards to reconstruction ability, we calculate the $L_1$ distance, peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) on CelebA. As for the identity-preserving ability, we calculate the cosine similarity of feature vectors corresponding to the input face images and the re-rendered face images, which are extracted by a pre-trained LightCNN-29 (Wu et al. 2018) and evoLVe model (Zhou et al. 2019). Comparison results are shown in Tab. 1. We can notice that our method greatly outperforms the baseline methods in both aspects. This suggests that our method has big potential in wide application scenarios.

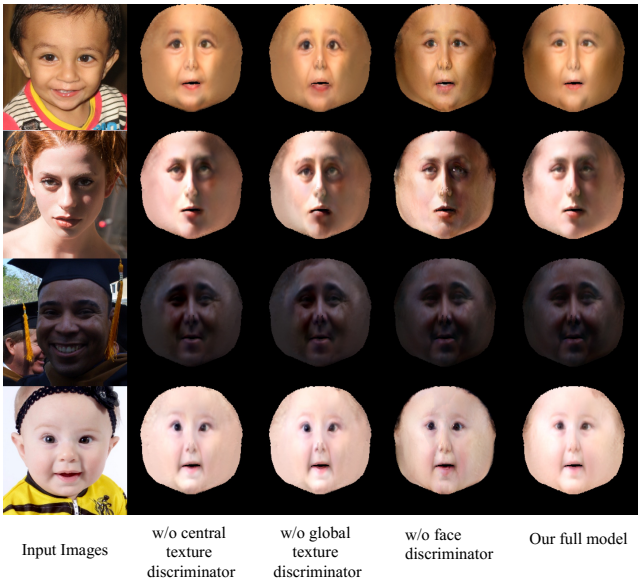


Figure 5: Qualitative comparisons of the results in ablation study.

| | | | | | |
|---|---|---|---|---|---|
| Losses | Reconstruction Loss | ✓ | ✓ | ✓ | ✓ |
| | Perceptual Loss | | ✓ | ✓ | ✓ |
| | Identity Loss | ✓ | | ✓ | ✓ |
| | Adversarial Loss | ✓ | ✓ | | ✓ |
| PSNR ↑ | | 28.10 | 27.83 | 29.34 | **30.78** |
| SSIM ↑ | | 0.831 | 0.817 | 0.865 | **0.897** |

Table 2: Quantitative comparisons of the results in ablation study.

## Ablation Study

We perform an ablation study of our method in terms of multi-view GAN inversion, and the two heads in discrimination. A qualitative comparison is shown in Fig. 5, from which we can see that the essential components in our method help improve the image quality.

# Conclusion

In this paper, we present a novel method (MvInvert) for face self-occlusion recovery. Our method produces a textured 3D face faithfully reconstructing the input portrait with good identity-preserving capability. Instead of using expensive paired image-texture data, we propose to leverage face priori encoded in a generative model to recover the incomplete UV textures with high-fidelity by using multi-view GAN inversion. In addition, we implement a CUDA-accelerated Poisson blending equation solver which largely reduces the running time of rendering. Quantitative and qualitative evaluations demonstrate that our method outperforms the state-of-art methods.

## Acknowledgements

This research was supported in part by the Natural Science Foundation of China (grants 61732004 and 62176249), and the Youth Innovation Promotion Association CAS (grant 2018135).

## References

Abdal, R.; Qin, Y.; and Wonka, P. 2019. Image2stylegan: How to embed images into the stylegan latent space? In *IEEE/CVF ICCV*, 4432–4441.

Abdal, R.; Qin, Y.; and Wonka, P. 2020. Image2stylegan++: How to edit the embedded images? In *IEEE/CVF CVPR*, 8296–8305.

Alaluf, Y.; Patashnik, O.; and Cohen-Or, D. 2021. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. *arXiv preprint arXiv:2104.02699*.

Arnaud Dessein, W. A. S.; and Richard C Wilson, E. R. H. 2014. Seamless texture stitching on a 3D mesh by poisson blending in patches. In *IEEE ICIP*, 2031–2035.

Blanz, V.; and Vetter, T. 1999. A morphable model for the synthesis of 3D faces. In *ACM SIGGRAPH*, 187–194.

Booth, J.; Antonakos, E.; Ploumpis, S.; Trigeorgis, G.; Panagakis, Y.; and Zafeiriou, S. 2017. 3D face morphable models. In *IEEE/CVF CVPR*, 5464–5473.

Booth, J.; Roussos, A.; Zafeiriou, S.; Ponniah, A.; and Dunaway, D. 2016. A 3d morphable model learnt from 10,000 faces. In *IEEE/CVF CVPR*, 5543–5552.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Bulat, A.; and Tzimiropoulos, G. 2017. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *IEEE ICCV*, 1021–1030.

Cao, C.; Weng, Y.; Zhou, S.; Tong, Y.; and Zhou, K. 2013. Facewarehouse: A 3d facial expression database for visual computing. *IEEE TVCG*, 20(3): 413–425.

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 *IEEE/CVF CVPR*, 7093–7102.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019a. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *IEEE/CVF CVPR*, 4690–4699.

Deng, Y.; Yang, J.; Xu, S.; Chen, D.; Jia, Y.; and Tong, X. 2019b. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE/CVF CVPRW*, 0–0.

Gecer, B.; Deng, J.; and Zafeiriou, S. 2021. OSTeC: One-Shot Texture Completion. In *IEEE/CVF CVPR*, 7628–7638.

Gecer, B.; Ploumpis, S.; Kotsia, I.; and Zafeiriou, S. 2019. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *IEEE/CVF CVPR*, 1155–1164.

Guo, Y.; Cai, J.; Jiang, B.; Zheng, J.; et al. 2018. Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE TPAMI*, 41(6): 1294–1307.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF CVPR*, 4401–4410.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *IEEE/CVF CVPR*, 8110–8119.

Koppen, P.; Feng, Z.-H.; Kittler, J.; Awais, M.; Christmas, W.; Wu, X.-J.; and Yin, H.-F. 2018. Gaussian mixture 3D morphable face model. *PR*, 74: 617–628.

Lee, C.-H.; Liu, Z.; Wu, L.; and Luo, P. 2020a. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *IEEE/CVF CVPR*, 5549–5558.

Lee, G.-H.; and Lee, S.-W. 2020. Uncertainty-aware mesh decoder for high fidelity 3d face reconstruction. In *IEEE/CVF CVPR*, 6100–6109.

Lee, M.; Cho, W.; Kim, M.; Inouye, D.; and Kwak, N. 2020b. Styleuv: Diverse and high-fidelity uv map generative model. *arXiv preprint arXiv:2011.12893*.

Lin, J.; Yuan, Y.; Shao, T.; and Zhou, K. 2020. Towards high-fidelity 3D face reconstruction from in-the-wild images using graph convolutional networks. In *IEEE/CVF CVPR*, 5891–5900.

Lin, J.; Yuan, Y.; and Zou, Z. 2021. MeInGame: Create a Game Character Face from a Single Portrait. In *AAAI*, 311–319.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2018. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018): 11.

Ma, F.; Ayaz, U.; and Karaman, S. 2019. Invertibility of convolutional generative networks from partial measurements. In *NIPS*, 9628–9637.

Richardson, E.; Alaluf, Y.; Patashnik, O.; Nitzan, Y.; Azar, Y.; Shapiro, S.; and Cohen-Or, D. 2021. Encoding in style: a stylegan encoder for image-to-image translation. In *IEEE/CVF CVPR*, 2287–2296.

Richardson, E.; Sela, M.; Or-El, R.; and Kimmel, R. 2017. Learning detailed face reconstruction from a single image. In *IEEE/CVF CVPR*, 1259–1268.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *IEEE/CVF CVPR*, 815–823.

Shang, J.; Shen, T.; Li, S.; Zhou, L.; Zhen, M.; Fang, T.; and Quan, L. 2020. Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency. In *ECCV 2020*, 53–70.

Shen, Y.; Gu, J.; Tang, X.; and Zhou, B. 2020. Interpreting the latent space of gans for semantic face editing. In *IEEE/CVF CVPR*, 9243–9252.

Smith, W. A.; Seck, A.; Dee, H.; Tiddeman, B.; Tenenbaum, J. B.; and Egger, B. 2020. A morphable face albedo model. In *IEEE/CVF CVPR*, 5011–5020.

Tov, O.; Alaluf, Y.; Nitzan, Y.; Patashnik, O.; and Cohen-Or, D. 2021. Designing an encoder for stylegan image manipulation. *ACM TOG*, 40(4): 1–14.

Tran, L.; and Liu, X. 2018. Nonlinear 3d face morphable model. In *IEEE/CVF CVPR*, 7346–7355.

Wu, X.; He, R.; Sun, Z.; and Tan, T. 2018. A light cnn for deep face representation with noisy labels. *IEEE TIFS*, 13(11): 2884–2896.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Learning Face Representation from Scratch. *arXiv preprint arXiv:1411.7923*.

Yin, L.; Wei, X.; Sun, Y.; Wang, J.; and Rosato, M. J. 2006. A 3D facial expression database for facial behavior research. In *FG*, 211–216.

Yin, X.; Huang, D.; Fu, Z.; Wang, Y.; and Chen, L. 2021. Weakly-Supervised Photo-realistic Texture Generation for 3D Face Reconstruction. *arXiv preprint arXiv:2106.08148*.

Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; and Sang, N. 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 325–341.

Yu, Y.; Gong, Z.; Zhong, P.; and Shan, J. 2017. Unsupervised representation learning with deep convolutional neural network for remote sensing images. In *ICIG*, 97–108.

Zhou, H.; Liu, J.; Liu, Z.; Liu, Y.; and Wang, X. 2020. Rotate-and-Render: Unsupervised Photorealistic Face Rotation From Single-View Images. In *IEEE/CVF CVPR*, 5911–5920.

Zhou, Y.; Deng, J.; Kotsia, I.; and Zafeiriou, S. 2019. Dense 3d face decoding over 2500fps: Joint texture & shape convolutional mesh decoders. In *IEEE/CVF CVPR*, 1097–1106.

Zhu, J.; Shen, Y.; Zhao, D.; and Zhou, B. 2020. In-domain gan inversion for real image editing. In *ECCV*, 592–608.

Zhu, J.-Y.; Krähenbühl, P.; Shechtman, E.; and Efros, A. A. 2016. Generative visual manipulation on the natural image manifold. In *ECCV*, 597–613.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2223–2232.

Zhu, J.-Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017b. Multimodal Image-to-Image Translation by Enforcing Bi-Cycle Consistency. In *NIPS*, 465–476.