Controllable 3D Face Synthesis with Conditional Generative Occupancy Fields

Keqiang Sun[1,*], **Shangzhe Wu**[2*], **Zhaoyang Huang**[1],
Ning Zhang[3], **Quan Wang**[3], **Hongsheng Li**[1,4]
[1]CUHK MMLab [2]Oxford VGG [3]SenseTime Research
[4]Centre for Perceptual and Interactive Intelligence Limited
kqsun@link.cuhk.edu.hk, szwu@robots.ox.ac.uk, hsli@ee.cuhk.edu.hk

Abstract

Capitalizing on the recent advances in image generation models, existing controllable face image synthesis methods are able to generate high-fidelity images with some levels of controllability, e.g., controlling the shapes, expressions, textures, and poses of the generated face images. However, these methods focus on 2D image generative models, which are prone to producing inconsistent face images under large expression and pose changes. In this paper, we propose a new NeRF-based conditional 3D face synthesis framework, which enables 3D controllability over the generated face images by imposing explicit 3D conditions from 3D face priors. At its core is a conditional Generative Occupancy Field (cGOF) that effectively enforces the shape of the generated face to commit to a given 3D Morphable Model (3DMM) mesh. To achieve accurate control over fine-grained 3D face shapes of the synthesized image, we additionally incorporate a 3D landmark loss as well as a volume warping loss into our synthesis algorithm. Experiments validate the effectiveness of the proposed method, which can generate high-fidelity face images and shows more precise 3D controllability than state-of-the-art 2D-based controllable face synthesis methods. Find code and more demo at https://keqiangsun.github.io/projects/cgof.

1 Introduction

Recent success of Generative Adversarial Networks (GANs) [13] has led to tremendous progress in face image synthesis. State-of-the-art methods, such as StyleGAN [21, 22, 20], are capable of generating photo-realistic face images. Apart from photo-realism, being able to control the appearance of the generated images is also key in many real-world applications, such as face animation, reenactment, and free-viewpoint rendering. Early works on controllable face synthesis rely on external attribute annotations to learn an attribute-guided face image generation model [27, 11, 55]. However, these attributes, such as "big nose", "chubby" and "smiling" in CelebA dataset [26], can only provide abstract semantic-level guidance on the generation, and the generated faces often lack 3D geometric consistency. Moreover, it is often much harder to obtain low-level geometric annotations beyond semantic labels for direct 3D supervision.

Recently, researchers have attempted to incorporate 3D priors from parametric face models, such as 3D Morphable Models (3DMMs) [2, 38], into StyleGAN-based synthesis models, allowing for more precise 3D control over the generated images, including facial expressions and head poses [8, 52, 39]. Despite their impressive image quality, these models still tend to produce *3D inconsistent* faces under large expression and pose variations due to the lack of a 3D representation, as shown in Fig. 1.

[*]Equal Contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

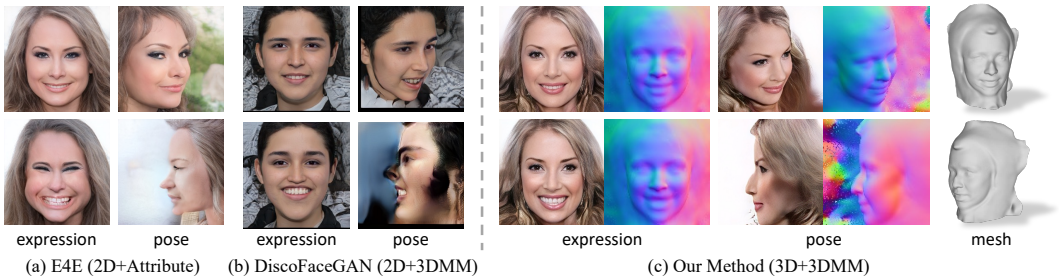


Figure 1: Existing controllable face synthesis methods, such as (a) E4E [53] and (b) DiscoFaceGAN [8], fail to preserve consistent geometry in the face images generated with large expression and pose variations, due to the lack of a 3D representation. (c) Our proposed *controllable 3D face synthesis* method leverages a NeRF-based 3D generative model conditioned on a prior 3DMM, which enables precise, disentangled 3D control over the generated face images.

With the advances in differentiable rendering and neural 3D representations, a recent line of work has explored photo-realistic 3D face generation using only 2D image collections as training data [49, 5, 45, 14, 37, 4, 57, 9]. Neural Radiance Fields (NeRFs) [31], in particular, have enabled 3D generative models, such as pi-GAN [5] and StyleNeRF [14], to synthesize high-fidelity, 3D consistent faces, by training with 2D images only. However, these models are purely generative and do not support precise 3D control over the generated faces, such as facial expressions.

In this work, we aim to connect these two groups of research—controllable face synthesis and 3D generative models—and present a NeRF-based 3D conditional face synthesis model that can generate high-fidelity face images with precise 3D control over the 3D shape, expression, and pose of the generated faces, by leveraging a parametric 3DMM face model. Imposing precise 3D conditions on an implicit neural radiance field is non-trivial. A naive baseline solution is to enforce that the input 3DMM parameters can be reproduced from the generated face image via a 3DMM reconstruction model, similar to [8]. Unfortunately, this 3DMM parameter reconstruction loss only provides *indirect* supervisory signals to the underlying NeRF volume, and is insufficient for achieving precise 3D control, as shown in Fig. 5.

We seek to impose explicit 3D condition *directly* on the NeRF volume. To this end, we propose a conditional Generative Occupancy Field (cGOF). It consists of a mesh-guided volume sampling procedure and a complimentary density regularizer that gradually concentrates the volume density around a given mesh. In order to achieve fine-grained 3D control, such as expression, we further introduce a 3D landmark loss and a volume warping loss.

The main contributions are as follows. We propose a novel controllable 3D face synthesis method that learns to generate high-fidelity face images with precise controllability over the 3D shape, expression, and poses, given only a collection of 2D images as training data. This is achieved by a novel conditional Generative Occupancy Field representation and a set of volumetric losses that effectively condition the generated NeRF volumes on a parametric 3D face model.

2 Related Work

Controllable Generative Adversarial Networks. Generative Adversarial Nets (GANs) [13] gained popularity over the last decade due to their remarkable image generation ability [42, 19, 21, 22]. Prior works have studied disentangled representation learning in generative models [16, 52, 51, 23, 46, 1, 32, 6, 7, 25]. Existing work on controllable facial image synthesis [40, 52, 51, 8, 39] relies on 2D image-based generative models, which does not guarantee 3D consistency in the generated images.

3D-Aware GAN. Another line of work in 3D-Aware generative models has looked into disentangling 3D geometric information from 2D images, and learning to generate various 3D structures, such as voxels [56, 60, 12, 32, 33, 28], meshes [49, 24], and Neural Radiance Fields (NeRFs) [5, 58, 14, 44, 36, 35], by training on image collections using differentiable rendering. The key idea is to use a discriminator that encourages images rendered from random viewpoints sampled from a prior pose distribution to be indistinguishable from real images. Among these works, pi-GAN has demonstrated high-fidelity 3D synthesis results adopting a NeRF representation. SofGAN [6]

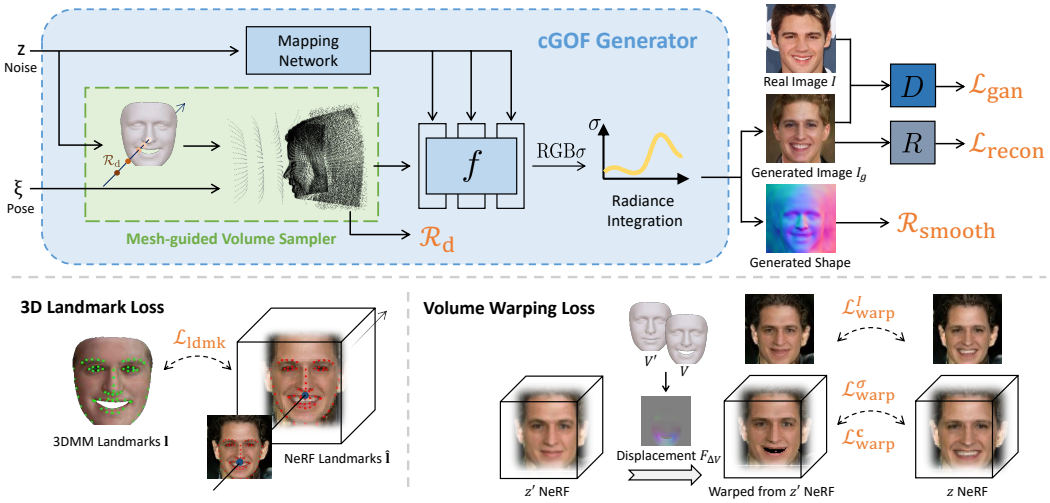


Figure 2: Method overview. *(Top)*: Conditional Generative Occupancy Field (cGOF) leverages a mesh-guided volume sampler which effectively conditions the generated NeRF on an input 3DMM mesh. It is trained in an adversarial learning framework using only single-view images. *(Bottom)*: The 3D landmark loss encourages the semantically important facial landmarks to follow the input mesh, and the volume warping loss enforces two NeRF volumes generated with different expression codes to be consistent through a warping field induced from the corresponding 3DMM meshes.

ensures viewpoint consistency via the proposed semantic occupancy field but requires 3D meshes with semantic annotations for pre-training. In this work, we build on top of pi-GAN and directly learn from unlabeled image collections with a prior 3DMM.

3D Priors for Face Generation. To learn controllable 3D face synthesis models, researchers have leveraged prior face models. 3D Morphable Models (3DMMs) [38, 2] is a widely-used parametric face model, which represents the human face as a set of PCA bases derived from 3D scans. Many previous works [8, 52, 6, 39] have proposed to incorporate 3D priors of 3DMM into generative models to enable controllable 3D face synthesis. In particular, DiscoFaceGAN [8], PIE [51] and StyleRig [52] leverage 3DMM to establish losses to ensure 3D consistency, but still use a 2D image-based StyleGAN generator. HeadNeRF [18] uses a 3D NeRF representation as well as 3DMM but is trained with a reconstruction loss using annotated multi-view datasets. Our method is trained on *unannotated single-view* images only, by directly imposing 3DMM conditions into 3D NeRF volume.

3 Method

Given a collection of single-view real-world face images $\mathcal{Y}$, where each face is assumed to appear only once without multiple views, the goal of this work is to learn a controllable 3D face synthesis model that can generate face images with desired 3D shape, expression, and pose. To achieve this, we propose a NeRF-based conditional 3D generative model G and incorporate priors from a parametric 3D Morphable Model (3DMM) [38]. At the core of our method is a novel conditional Generative Occupancy Field (cGOF) as well as two auxiliary volumetric losses that enable effective 3D control based on the 3DMM, as illustrated in Figure 2.

3.1 Preliminaries on NeRF and pi-GAN

Neural Radiance Field (NeRF). NeRF [31] represents a 3D scene as a radiance field parametrized by a Multi-Layer Perceptron (MLP) that predicts a density value $\sigma \in \mathbb{R}$ and a radiance color $\mathbf{c} \in \mathbb{R}^3$ for every 3D location, given its xyz coordinates and the viewing direction as input. To render a 2D image of the scene from an arbitrary camera viewpoint ξ, we cast rays through the pixels into the 3D volume, and evaluate the density values and radiance colors at a number of sampled points along each ray. Specifically, for each pixel, the camera ray $\mathbf{r}$ can be expressed as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o}$ denotes the camera origin, $\mathbf{d}$ the direction, and t defines a sample point within near and far bounds t_{n} and t_{f}. For each 3D sample point $\mathbf{x}_i$, we query the NeRF network f to obtain its σ_i and radiance color $\mathbf{c}_i$:

$(\sigma_i, \mathbf{c}_i) = f(\mathbf{x}_i, \mathbf{d})$. Using volume rendering [29, 31], the final color of the pixel is given by

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i \,, \quad \text{where} \quad T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \,, \tag{1}$$

and $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples. To optimize a NeRF on a 3D scene, the photometric loss is computed between the rendered pixels and the ground-truth images captured from a dense set of viewpoints.

Generative Radiance Field. Several follow-up works have extended this representation to a generative framework [5, 14, 9, 57]. Instead of using a multi-view reconstruction loss, they train a discriminator that encourages images rendered from randomly sampled viewpoints to be indistinguishable from real images, allowing the model to be trained with only single-view image collections.

Our method is built on top of pi-GAN [5], a generative radiance field model based on a SIREN-modulated NeRF representation [47]. During training, images are rendered from random viewpoints ξ sampled from a predefined pose distribution p_ξ (estimated from the training dataset). To model geometry and appearance variations, a random noise code $\mathbf{z}$ is sampled from a standard normal distribution and mapped to a number of frequencies γ_i and phase shifts β_i through a mapping network. These encodings are used to modulate the outputs of each layer in the NeRF MLP f before the sinusoidal activations. The rendered images $I_{\text{g}} = G(\mathbf{z}, \xi)$ and the real images I randomly sampled from the training dataset are then passed into the discriminator D, and the model is trained using a non-saturating GAN loss with R1 regularization following [30]:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{z}\sim p_z, \xi\sim p_\xi}[f(D(G(\mathbf{z}, \xi)))] + \mathbb{E}_{I\sim p_{\mathcal{D}}}[f(-D(I)) + \lambda |\nabla D(I)|^2], \tag{2}$$

where $f(u) = -\log(1 + \exp(-u))$ and λ is a hyperparameter balancing the regularization term.

3.2 Controllable 3D Face Synthesis

In order to learn a controllable 3D face synthesis model, we leverage priors of a 3DMM [38], which is a parametric morphable face model, where the shape, expression, texture, and other factors of a face are modeled by a set of PCA bases and coefficients $\tilde{\mathbf{z}} = (\tilde{\mathbf{z}}_{\text{shape}}, \tilde{\mathbf{z}}_{\text{exp}}, \tilde{\mathbf{z}}_{\text{tex}}, \tilde{\mathbf{z}}_{\text{else}})$ respectively. Our goal is to condition the generated face image I_{g} on a set of 3DMM parameters $\tilde{\mathbf{z}}$ as well as a camera pose ξ, which specifies the desired configuration of the generated face image.

Baseline. To enforce such conditioning, following [8], we make use of an off-the-shelf pre-trained and fixed 3DMM reconstruction model R that predicts a set of 3DMM parameters from a single 2D image [10]. This allows us to incorporate a 3DMM parameter reconstruction loss that ensures the generated face images commit to the input condition.

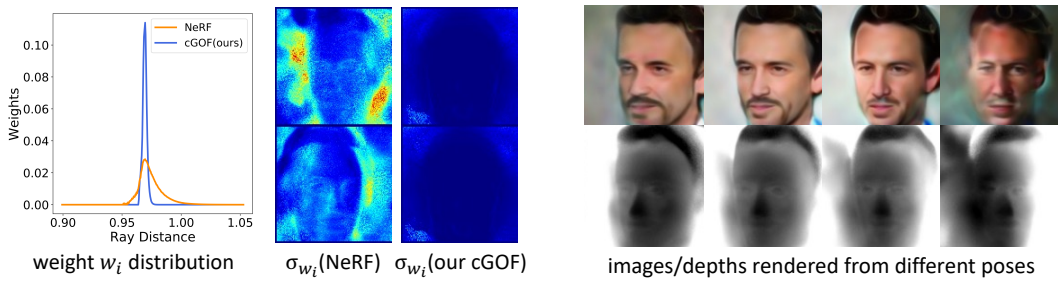
Specifically, we sample a random noise $\mathbf{z} \in \mathbb{R}^d$ from a standard normal distribution $p_{\mathbf{z}}$, which is later passed into the pi-GAN generator G to produce a face image I_{g}. Unlike [8], which additionally trains a set of VAEs to map the noise $\mathbf{z}$ to meaningful 3DMM coefficients $\tilde{\mathbf{z}}$, we assume that the 3DMM coefficients of all the training images follow a normal distribution and simply standardize them to obtain the corresponding normalized noise code $\mathbf{z} = \tau(\tilde{\mathbf{z}}) = \tilde{L}^{-1}(\tilde{\mathbf{z}} - \tilde{\mu})$, where $\tilde{\mu}$ and $\tilde{L}$ are the mean and Cholesky decomposition component of covariance matrix of the 3DMM coefficients estimated from all the training images using the 3DMM reconstruction model R.

During training, we predict 3DMM parameters from the generated image using the reconstruction model R, normalize it and minimize a reconstruction loss between the predicted normalized parameters and the input noise:

$$\mathcal{L}_{\text{recon}} = \|\hat{\mathbf{z}} - \mathbf{z}\|_1, \quad \text{where} \quad \hat{\mathbf{z}} = \tau(R(G(\mathbf{z}, \xi))). \tag{3}$$

This results in a baseline model, where the conditioning is implicitly imposed through the 3DMM parameter reconstruction loss. Example results of this baseline model can be seen in Fig. 5.

Challenges. There are several issues with this baseline model. First, the effectiveness of the conditioning depends heavily on the accuracy and generalization performance of the 3DMM reconstruction model R. Fundamentally, these parameters only describe an extremely abstract, semantic representation of human faces, which is insufficient to govern the fine-grained geometric details in real faces precisely. Since our model generates a 3D representation for rendering, a more effective solution is



(a) Weight Distribution Comparison

(b) Shape-Texture Ambiguity w/o cGOF

Figure 3: Challenges in imposing 3D condition on NeRF. (a) Without the proposed cGOF (mesh-guided sampler and density regularizer), directly supervising the over-parameterized NeRF volume with a depth loss leads to ambiguous surfaces indicated by high variance of the rendering weights σ_{w_i}. (b) Rendering semi-transparent volumes leads to poor realism and multi-view consistency.

to impose the conditioning directly on the 3D volume, instead of just on the rendered 2D images. However, the radiance field representation is highly over-parametrized [59]. The color of a pixel is computed by integrating the radiance along the entire camera ray, whereas in many cases in reality, it should only reflect the color of the intersection point on an opaque surface. Incorporating losses directly on this over-parametrized 3D volume (*e.g.* a depth loss) tends to hamper its converging to an opaque surface, resulting in a semi-transparent volume that no longer guarantees multi-view consistency, as shown in Fig. 3.

3.3 Conditional Generative Occupancy Field

In order to impose precise 3D control on the NeRF volume, we first identify an effective way of enforcing the volume density to converge to the surface of human faces. Inspired by [57], we propose conditional Generative Occupancy Fields (cGOF), which consists of a mesh-guided volume sampler (MgS) and a distance-aware volume density regularizer.

The key idea is to use the conditioning 3DMM mesh to explicitly guide the volume sampling procedure. Given a random noise $\mathbf{z}$, we first obtain the corresponding 3DMM coefficients $\tilde{\mathbf{z}} = \tau^{-1}(\mathbf{z})$ by denormalization, which consists of $(\tilde{\mathbf{z}}_{\text{shape}}, \tilde{\mathbf{z}}_{\text{exp}}, \tilde{\mathbf{z}}_{\text{tex}}, \tilde{\mathbf{z}}_{\text{else}})$. With these coefficients, we can sample a 3D face mesh M_{in} from the 3DMM Λ:

$$M_{\text{in}} = \Lambda(\mathbf{z}_{\text{shape}}, \mathbf{z}_{\text{exp}}) = M_{\text{mean}} + B_{\text{shape}} \cdot \tilde{\mathbf{z}}_{\text{shape}} + B_{\text{exp}} \cdot \tilde{\mathbf{z}}_{\text{exp}}, \tag{4}$$

where M_{mean} is the mean shape, and B_{shape} and B_{exp} are the deformation bases for shape and expression in 3DMM respectively. During rendering, this mesh will serve as a 3D condition, allowing us to apply importance sampling around the surface region, and suppress the volume densities far away from the surface.

Specifically, for each pixel within the face region, we find its distance t_{m} to the input 3DMM mesh M_{in} along the ray direction by rendering a depth map, taking into account the perspective projection. This allows us to sample points along the ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ within a set of uniformly spaced intervals around the surface intersection point of the ray and the input mesh:

$$t_i \sim \mathcal{U}\left[t_{\text{m}} + (\frac{i-1}{N} - \frac{1}{2})\delta, t_{\text{m}} + (\frac{i}{N} - \frac{1}{2})\delta\right], \text{ where } i = 1, 2, \ldots, N_{\text{surf}}. \tag{5}$$

where N_{surf} is the number of sampled points, and δ is a parameter controlling the "thickness" of the sampling region. We gradually shrink this margin from 0.5 to 0.05 of the volume range.

Furthermore, we suppress the volume density in the space far away from the given mesh using a distance-aware volume density regularizer. Specifically, we additionally sample a sparse set of points along each ray, and denote the absolute distances from these sampling points to the mesh surface as $d_i = |t_i - t_{\text{m}}|, i = 1, \ldots, N_{\text{vol}}$. The distance-aware volume density regularizer R_{d} is defined as:

$$R_{\text{d}} = \sum_{i=0}^{N_{\text{vol}}} \sigma_i \cdot \left[\exp\left(\alpha \cdot \max(d_i - \delta/2, 0)\right) - 1\right], \tag{6}$$

where σ_i is the volume density evaluated at those sample points, and α is an inverse temperature parameter (set to 20), which controls the strength of the penalty response to increased distance Δd_i.

Note that this penalty only applies to sample points beyond the surface sampling range $\frac{\delta}{2}$. The mesh-guided volume sampler together with the volume density regularizer effectively constrains the generated NeRF volume to follow closely the input 3DMM mesh, as shown in Tab. 2.

3.4 Auxiliary Losses for Fine-grained 3D Control

So far, the model is able to generate 3D faces following the overall shape of the conditioning 3DMM mesh. However, since 3DMM is only a low-dimensional PCA model of human faces without capturing the fine-grained shape and texture details of real faces, enforcing the generated 3D faces to fully commit to the coarse 3DMM meshes actually leads to over-smoothed shapes, and does not guarantee precise control over the geometric details, especially under large facial expression changes. To enable precise geometric control, we further introduce two auxiliary loss terms.

3D Landmark Loss encourages the semantically important facial landmarks to closely follow the input condition. Specifically, we identify a set of 3DMM vertices that correspond to $N_{\text{k}} = 68$ facial landmarks [43, 10]. Let $\{\mathbf{l}_k\}_{k=1}^{N_{\text{k}}}$ denote the 3D locations of these landmarks on the input 3DMM mesh. In order to find the same set of landmarks on the generated image, we extract 3D landmarks $\{\hat{\mathbf{l}}_k\}_{k=1}^{N_{\text{k}}}$ from the *estimated* 3DMM mesh and project them onto the 2D image to obtain the 2D projections. We then render the depth values of NeRF at these 2D locations using volume rendering and back-project them to 3D to obtain the 3D landmark locations of the generated volume $\{\hat{\mathbf{l}}'_k\}$. To avoid the impact of occlusion, we skip the facial contour landmarks (numbered from 1 to 17) when calculating loss. The landmark loss is defined among these three sets of 3D landmarks:

$$\mathcal{L}_{\text{ldmk}} = \sum_{k=1}^{N_{\text{k}}} \|\hat{\mathbf{l}}_k - \mathbf{l}_k\|_1 + \sum_{k=18}^{N_{\text{k}}} \|\hat{\mathbf{l}}'_k - \mathbf{l}_k\|_1. \tag{7}$$

Adopting this 3D landmark loss further encourages the reconstructed 3D faces to align with the input mesh at these semantically important landmark locations, which is particularly helpful in achieving precise expression control, as validated in Tab. 2.

Volume Warping Loss enforces that the NeRF volume generated with a different expression code should be consistent with the original volume warped by a warping field induced from the two 3DMM meshes. More concretely, let $M = \Lambda(\mathbf{z}_{\text{shape}}, \mathbf{z}_{\text{exp}})$ be a 3DMM mesh randomly sampled during training, $M' = \Lambda(\mathbf{z}_{\text{shape}}, \mathbf{z}'_{\text{exp}})$ be a mesh of the same face but with a different expression code $\mathbf{z}'_{\text{exp}}$, and V and V' be their vertices respectively. We first compute the vertex displacements denoted by $\Delta V = V' - V$, and render them into a 2D displacement map $F_{\Delta V} \in \mathbb{R}^{H \times W \times 3}$, where each pixel $F_{\Delta V}^{(u,v)}$ defines the 3D displacement of its intersection point on the mesh M.

For each pixel (u, v) in the image I_{g} generated from the first NeRF, we obtain the densities and radiance colors $(\sigma_i, \mathbf{c}_i) = f(\mathbf{x}_i, \mathbf{d}, \mathbf{z})$ of the N_{surf} mesh-guided sample points $\mathbf{x}_i$ described above. The superscript (u, v) is dropped for simplicity. We warp all the 3D sample points along each ray using the same 3D displacement value rendered at the pixel $F_{\Delta V}$ to obtain the warped 3D points $\mathbf{x}'_i = \mathbf{x}_i + F_{\Delta V}$ (assuming that all the points are already close to the mesh surface). We then query the second NeRF generated with $\mathbf{z}'_{\text{exp}}$ at these warped 3D locations $\mathbf{x}'_i$ to obtain another set of densities and radiance colors $(\sigma'_i, \mathbf{c}'_i) = f(\mathbf{x}'_i, \mathbf{d}', \mathbf{z}')$, and encourage them to stay close to the original ones.

In addition, we also enforce the images rendered from the two radiance fields to be consistent. We integrate the radiance colors $\mathbf{c}'_i$ of the *warped* points with their densities σ'_i using Eq. (1), and obtain an image $\hat{I}_{\text{g}}$ (with all pixels). This image $\hat{I}_{\text{g}}$ rendered from the second radiance field is encouraged to be identical to the original image I_{g}. The final warping loss is thus composed of three terms:

$$\mathcal{L}_{\text{warp}} = \beta_{\text{d}} \cdot \sum_{i}^{N_{\text{surf}}} \|\sigma'_i - \sigma_i\|_1 + \beta_{\text{c}} \cdot \sum_{i}^{N_{\text{surf}}} \|\mathbf{c}'_i - \mathbf{c}_i\|_1 + \beta_{\text{I}} \cdot \|\hat{I}_{\text{g}} - I_{\text{g}}\|_1, \tag{8}$$

where β_{d}, β_{c} and β_{I} are the balancing weights.

3.5 Neural Renderer and Overall Training Objective

Due to the computation- and memory-expensive volume rendering process, the quality of the generated images is bound to a small resolution (64×64) in training. To improve the image quality without introducing too much computational overhead, we leverage an auxiliary neural renderer

Method Name	Venue	$DS_{\text{s}} \uparrow$	$DS_{\text{e}} \uparrow$	$DS_{\text{p}} \uparrow$	FID (512)$\downarrow$
PIE [51]	TOG'20	1.66	15.24	2.65	59.63
StyleRig [52]	CVPR'20	1.64	13.03	2.01	56.59
DiscoFaceGAN [8]	CVPR'20	5.97	15.70	5.23	76.13
E4E [53]	SIGGRAPH'21	1.91	8.66	7.08	96.40
Gan-Control [46]	ICCV'21	7.07	7.51	9.33	83.59
HeadNeRF [18]	CVPR'22	6.39	5.99	10.26	135.03
Ours w/o Neural Renderer	-	**23.24**	**29.13**	**23.45**	71.60
Ours w/ Neural Renderer	-	21.72	27.47	22.82	**31.83**

Table 1: DS and FID comparison with state-of-the-art controllable face synthesis methods.

introduced in [54] as a standalone module, which takes in the rendered coarse 2D image and produces a higher resolution version of it (512×512). We take their pre-trained model on the FFHQ dataset and finetune it on the CelebA-HQ dataset. This neural renderer significantly improves the quality of the generated images without compromising much 3D controllability, as shown in Tab. 1.

Similar to RegNeRF [34], we further impose a smoothness regularizer R_{smooth} on the generated NeRF volume by penalizing the L2 differences of neighboring pixels in both the depth maps and normal maps (referred to as $R^{\text{depth}}_{\text{smooth}}$ and $R^{\text{norm}}_{\text{smooth}}$ below). The final training objective is thus given by

$$\mathcal{L} = \lambda_{\text{gan}}\mathcal{L}_{\text{gan}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{d}}R_{\text{d}} + \lambda_{\text{ldmk}}\mathcal{L}_{\text{ldmk}} + \lambda_{\text{warp}}\mathcal{L}_{\text{warp}} + \lambda_{\text{smooth}}R_{\text{smooth}}, \tag{9}$$

where the λ's specify the weights for each term.

4 Experiments

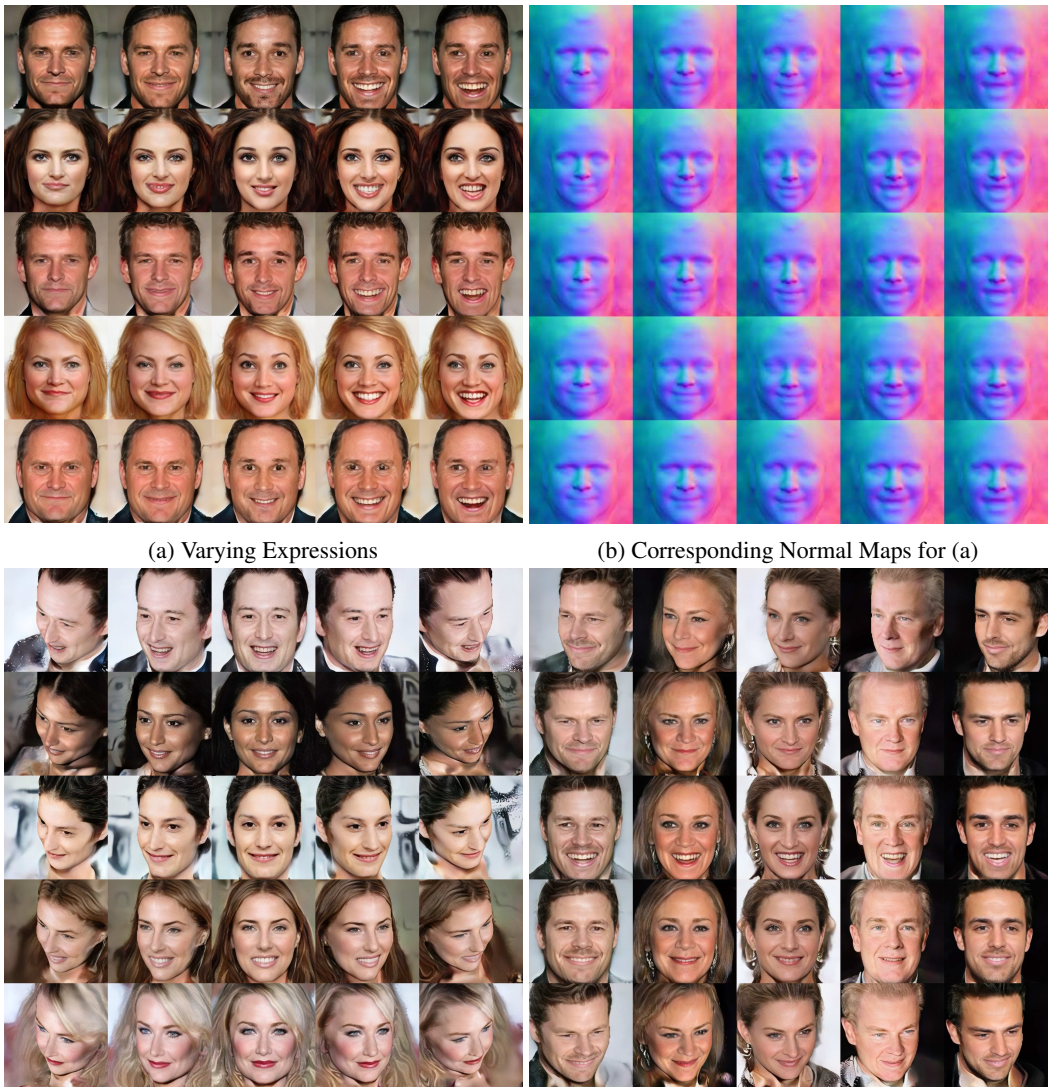
Datasets. We train our model on the CelebA [26] dataset, which consists of 200k celebrity face images with various poses, expressions, and lighting conditions. We use the cropped images and *do not make use of their facial attribute annotations*. The neural renderer module was originally trained on FFHQ dataset, which uses a slightly different cropping scheme. Therefore, we finetune their pre-trained model on the CelebA-HQ subset of CelebA containing 30k high-res images.

Implementation Details. We provide implementation details in the supplementary material, including network architectures and hyper-parameters. Here, we highlight a few important ones. We build our model on top of the implementation of pi-GAN [5], which uses a SIREN-based NeRF representation. We sample $N_{\text{surf}} = 12$ points around the 3DMM input mesh and $N_{\text{vol}} = 12$ coarse points for the density regularizer. The final model is trained for 72 hours on 8 GeForce GTX TITAN X GPUs.

Metrics. ***(1) Disentanglement Score (DS)***. Following [8], we evaluate 3D controllability using a 3DMM Disentanglement Score. To measure the DS on one disentangled property $\mathbf{z}_i$ (*e.g.* shape, expression or pose), we randomly sample a set of $\mathbf{z}_i$ from a normal distribution while keeping other codes fixed $\{\mathbf{z}_j\}, j \neq i$, and render images. We re-estimate the 3DMM parameters $\hat{\mathbf{z}}_i$ and $\{\hat{\mathbf{z}}_j\}$ from these rendered images and compute $DS_i = \prod_{j,j\neq i} \sigma_{\hat{\mathbf{z}}_i}/\sigma_{\hat{\mathbf{z}}_j}$, where $\sigma_{\hat{\mathbf{z}}_i}$ and $\sigma_{\hat{\mathbf{z}}_j}$ denote variance of the reconstructed parameters of the measured property $\hat{\mathbf{z}}_i$ and the rest $\hat{\mathbf{z}}_j$, respectively. A higher DS score indicates that the model achieves more precise controllability on that property while keeping the others unchanged. ***(2) Chamfer Distance (CD)***. We report the bi-directional Chamfer Distance between the generated face surface and the input 3DMM to measure how closely the generated 3D face follows the condition. ***(3) Landmark Distance (LD).*** To evaluate the fine-grained controllability of our model, we detect the 3D landmarks in the generated images using [10] and compute the Euclidean distance between the detected landmarks and the ones extracted from the input 3DMM vertices. ***(4) Landmark Correlation (LC).*** We measure the precision of expression control, by computing the 3D landmark displacements of two faces generated with a pair of random expression codes, and reporting the correlation score between these displacements and those obtained from the two 3DMM meshes. ***(5) Frechet Inception Distance (FID)***. The quality of the generated face images is measured using Frechet Inception Distance [17] with an ImageNet-pretrained Inception-V3 [50] feature extractor. We align images with 68 landmarks [43, 48, 41] to avoid domain shift.

4.1 Qualitative Evaluation

Fig. 4 shows some example images generated by our proposed cGOF model. In particular, we visualize the generated faces by varying only one of the controlled properties at a time, including



(a) Varying Expressions (b) Corresponding Normal Maps for (a)

(c) Varying Poses (d) Varying Identities

Figure 4: Face images generated by the proposed cGOF. The variations in the expression, pose and identity of the faces are highly disentangled and can be precisely controlled.

facial expression, head pose, and identity–combining shape and texture as in [8]. Our model is able to generate high-fidelity 3D faces with precise 3D controllability in a highly disentangled manner, even in the presence of large expression and pose variations. In contrast, existing methods tend to produce inconsistent face images as shown in Fig. 1. Please refer to the supplementary materials for more qualitative results and side-by-side comparisons against state-of-the-art methods.

4.2 Quantitative Evaluation

We compare our method to several state-of-the-art methods. E4E [53] and GAN-Control [46] are purely 2D controllable face image synthesis methods, which are based on facial attribute annotations. DiscoFaceGAN [8], PIE [51] and StyleRig [52] leverage 3DMM priors but still use a 2D image-based StyleGAN generator. HeadNeRF [18] uses a 3D NeRF representation as well as 3DMM but is trained with a reconstruction loss using annotated datasets, whereas our method is a 3D generative model trained on *unannotated single-view* images only.

We compare the controllability accuracy using the Disentanglement Score (see Col 3 to 5 of Tab. 1). Our method achieves significantly better disentanglement and more precise control in terms of shapes,

Index	Loss	CD ↓	LD ↓	LC ↑	DS_{s} ↑	DS_{e} ↑	DS_{p} ↑	FID (128) ↓
1	$\mathcal{L}_{\text{gan}}$	1.09	5.04	2.04	2.13	2.54	7.16	**18.76**
2	+ $\mathcal{L}_{\text{recon}}$(baseline)	0.87	3.85	26.15	3.56	5.03	11.00	21.97
3	(+ $\mathcal{L}_{depth}$)*	1.65	6.16	0.06	0.93	1.34	1.09	71.67
4	+ MgS	0.29	3.98	27.45	3.55	4.90	9.48	38.91
5	+ $\mathcal{R}_{\text{d}}$	0.31	3.51	51.74	3.56	5.31	15.94	29.62
6	+ $\mathcal{R}^{\text{norm}}_{\text{smooth}}$	0.27	4.72	30.25	3.18	4.43	16.26	31.63
7	+ $\mathcal{L}_{\text{ldmk}}$	0.39	1.86	84.43	16.54	16.65	21.24	56.90
8	+ $\mathcal{L}_{\text{warp}}$	**0.26**	1.44	89.91	20.47	22.04	22.91	47.18
9	+ $\mathcal{R}^{\text{depth}}_{\text{smooth}}$	0.27	**1.26**	**92.88**	**23.24**	**29.13**	**23.45**	26.64

Table 2: Ablation Study. Each row adds an additional loss term into the objective function upon all the loss terms above it. The last row is our final model. *Note that we do not use the depth loss in the final model, which compares the depths rendered from NeRF and 3DMM.

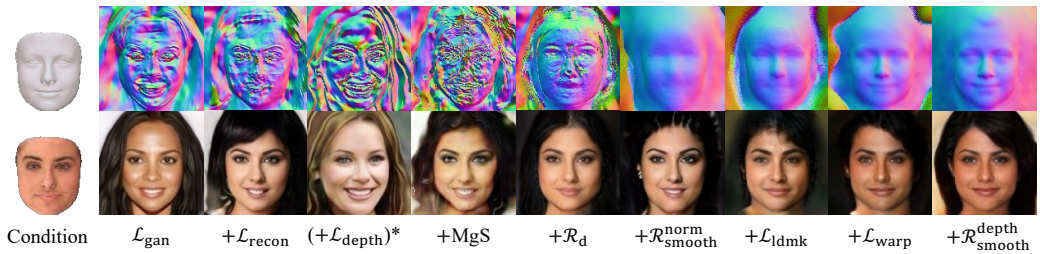


Figure 5: Ablation Study. We analyze the effects of the individual components and loss functions. With the components gradually enabled, the model is able to generate 3D realistic faces precisely following the condition 3DMM. Refer to Sec. 4.3 for more details. *Note that we do not use the depth loss in the final model, which compares the depths rendered from NeRF and 3DMM.

expressions, and poses, compared to existing methods. Note that since PIE and StyleRig do not have open-sourced code, we compute the score on the 167 generated images provided on the PIE project page[2]. We also compare the FID scores in Col 6 of Tab. 1. Our model achieves plausible image quality without the Neural Renderer, and outperforms the existing methods with the Neural Renderer.

4.3 Ablation Study

We conduct ablation studies to validate the effectiveness of each component in our model by adding them one by one, and evaluating the generated images with the four metrics specified above. Tab. 2 and Fig. 5 illustrate the studies and show that all the components improve the precision of 3D controllable image generation. In particular, the mesh-guided volume sampler (MgS) significantly improves the effectiveness of the 3DMM condition (denoted as '+ MgS' in Tab. 2). The 3D landmark loss $\mathcal{L}_{\text{ldmk}}$ (denoted as '+ $\mathcal{L}_{\text{ldmk}}$') and the volume warping loss $\mathcal{L}_{\text{warp}}$ (denoted as '+ $\mathcal{L}_{\text{warp}}$') further enhance the control over fine-grained geometric details. We also report FIDs of the outputs of the cGOF with the image resolution of 128×128. Overall, although the geometric regularizers lead to slightly degraded image quality (indicated by increased FIDs), they improve the 3D controllability by large margins (indicated by the improvement in other metrics).

5 Conclusions

We have presented a controllable 3D face synthesis method that can generate high-fidelity face images based on a conditional 3DMM mesh. We propose a novel conditional Generative Occupancy Field representation and a set of 3D losses that effectively impose 3D conditions directly on the generated NeRF volume, which enables much more precise 3D controllability over 3D shape, expression, and pose of the generated faces than state-of-the-art 2D counterparts. **Limitations.** The proposed method still lacks precise texture and illumination control over the generated face images. It also relies heavily on the prior 3DMM and does not generate high-fidelity shape details, such as wrinkles and hair. **Societal Impacts.** The proposed work might be used to generate fake facial images for imposture or

[2] `https://vcai.mpi-inf.mpg.de/projects/PIE/`

to create synthesized facial images to hide the true identities of cybercriminals. Defensive algorithms for recognizing such synthesized facial images should be accordingly developed.

6 Acknowledgements

This work is supported in part by Centre for Perceptual and Interactive Intelligence Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants (Nos. 14204021, 14207319).

We would like to thank Yu Deng for providing the evaluation code of the Disentanglement Score, and Jianzhu Guo for providing the pre-trained face reconstruction model. We are also indebted to thank Xingang Pan, Han Zhou, KwanYee Lin, Jingtan Piao, and Hang Zhou for their insightful discussions.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, pages 8296–8305, 2020.

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194, 1999.

[3] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *TVCG*, 20(3):413–425, 2013.

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

[5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, pages 5799–5809, 2021.

[6] Anpei Chen, Ruiyang Liu, Ling Xie, Zhang Chen, Hao Su, and Jingyi Yu. Sofgan: A portrait image generator with dynamic styling. *TOG*, 2021.

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, pages 2180–2188, 2016.

[8] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *CVPR*, pages 5154–5163, 2020.

[9] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. *CVPR*, 2022.

[10] 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 *CVPR Workshop*, 2019.

[11] Xing Di and Vishal M Patel. Face synthesis from visual attributes via sketch using conditional vaes and gans. *arXiv preprint arXiv:1801.00077*, 2017.

[12] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *3DV*, 2017.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 27, 2014.

[14] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *ICLR*, 2022.

[15] Yudong Guo, Jianfei Cai, Boyi Jiang, Jianmin Zheng, et al. Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *TPAMI*, 41(6):1294–1307, 2018.

[16] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. In *NeurIPS*, 2017.

[18] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. *CVPR*, 2022.

[19] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

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

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

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

[23] Thomas Leimkühler and George Drettakis. Freestylegan: Free-view editable portrait rendering with the camera manifold. *TOG*, 40(6), dec 2021.

[24] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *CVPR*, 2020.

[25] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *ICML*, pages 6127–6139, 2020.

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

[27] Yongyi Lu, Yu-Wing Tai, and Chi-Keung Tang. Attribute-guided face generation using conditional cyclegan. In *ECCV*, 2018.

[28] Sebastian Lunz, Yingzhen Li, Andrew Fitzgibbon, and Nate Kushman. Inverse graphics gan: Learning to generate 3d shapes from unstructured 2d data. *NeurIPS Workshop*, 2020.

[29] Nelson Max. Optical models for direct volume rendering. *TVCG*, 1(2):99–108, 1995.

[30] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *ICML*, 2018.

[31] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

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

[33] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *NeurIPS*, 2020.

[34] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022.

[35] Michael Niemeyer and Andreas Geiger. Campari: Camera-aware decomposed generative neural radiance fields. *3DV*, 2021.

[36] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021.

[37] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In *CVPR*, pages 13503–13513, June 2022.

[38] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *AVSS*, pages 296–301, 2009.

[39] Jingtan Piao, Keqiang Sun, Quan Wang, Kwan-Yee Lin, and Hongsheng Li. Inverting generative adversarial renderer for face reconstruction. In *CVPR*, pages 15619–15628, 2021.

[40] Shengju Qian, Kwan-Yee Lin, Wayne Wu, Yangxiaokang Liu, Quan Wang, Fumin Shen, Chen Qian, and Ran He. Make a face: Towards arbitrary high fidelity face manipulation. In *ICCV*, pages 10033–10042, 2019.

[41] Shengju Qian, Keqiang Sun, Wayne Wu, Chen Qian, and Jiaya Jia. Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation. In *ICCV*, pages 10153–10163, 2019.

[42] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[43] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV Workshop*, pages 397–403, 2013.

[44] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *NeurIPS*, 33:20154–20166, 2020.

[45] Yichun Shi, Divyansh Aggarwal, and Anil K. Jain. Lifting 2d stylegan for 3d-aware face generation. In *CVPR*, 2021.

[46] Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gerard Medioni. Gan-control: Explicitly controllable gans. In *ICCV*, pages 14083–14093, 2021.

[47] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020.

[48] 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 *ICCV*, pages 5462–5471, 2019.

[49] Attila Szabó, Givi Meishvili, and Paolo Favaro. Unsupervised generative 3d shape learning from natural images. *arXiv preprint arXiv:1910.00287*, 2019.

[50] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, June 2016.

[51] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. *TOG*, 39(6):1–14, 2020.

[52] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, pages 6142–6151, 2020.

[53] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *SIGGRAPH*, 2021.

[54] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *CVPR*, 2021.

[55] Yaohui Wang, Antitza Dantcheva, and Francois Bremond. From attribute-labels to faces: face generation using a conditional generative adversarial network. In *ECCV Workshop*, pages 0–0, 2018.

[56] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NeurIPS*, 2016.

[57] Xudong Xu, Xingang Pan, Dahua Lin, and Bo Dai. Generative occupancy fields for 3d surface-aware image synthesis. *NeurIPS*, 34, 2021.

[58] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *CVPR*, 2022.

[59] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020.

[60] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Joshua B. Tenenbaum, and William T. Freeman. Visual object networks: Image generation with disentangled 3D representations. In *NeurIPS*, 2018.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Sec. 6.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Sec. 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Sec. 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sec. 3
   (b) Did you include complete proofs of all theoretical results? [Yes] See Sec. 3
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We release the code and instructions at https://keqiangsun.github.io/projects/cgof.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 4 and the supplementary materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Sec. 4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] The license of the assets will be released together with our code.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We are using only public, standard benchmark datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing or conduct research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendices

A Implementation Details

Network Architectures. This project is based on the released code of pi-GAN [5] and Deep3DFaceRecon [10]. We use the same architectures for the generator and discriminator, as well as the main training process from the pi-GAN. To implement our proposed conditional Generative Occupancy Field (cGOF), we integrate a deep 3D face reconstruction model Deep3DFaceRecon [10] to reconstruct 3DMM parameters from the generated images for the 3DMM reconstruction parameter loss $\mathcal{L}_{\text{recon}}$. We follow [10] to adopt the popular 2009 Basel Face Model [38] for shape and texture bases, and use the expression bases of [15], built from FaceWarehouse [3]. The expression components are the PCA model of the offsets between the expression meshes and the neutral meshes of individual persons.

Coordinate System Alignment. We align the 3DMM meshes to pi-GAN to impose the 3D losses. We first use the pre-trained pi-GAN to generate multi-view images of various face instances in the canonical viewpoint and use the Deep3DFaceRecon to reconstruct 3DMM meshes M for the generated faces, on which we define a set of landmarks $\mathbf{l}_{\text{3D}}$. Let K_{recon} and E_{recon} be the intrinsic and extrinsic matrices for 3DMM, and K_{pigan} and E_{pigan} the counterparts for pi-GAN respectively. Let T_{R2P} be the transformation matrix from 3DMM to pi-GAN that we would like to estimate. We directly optimize this transformation matrix by minimizing the difference of 2D projections of a set of landmarks obtained from the original mesh M and the transformed mesh $M' = T_{\text{R2P}} \cdot M$, denoted as $\mathbf{l}_{\text{2D}}$ and $\mathbf{l}'_{\text{2D}}$:

$$\begin{aligned} T^*_{\text{R2P}} &= \arg\min_{T_{\text{R2P}}} \|\mathbf{l}_{\text{2D}} - \mathbf{l}'_{\text{2D}}\|_1, \\ \text{where} \quad \mathbf{l}_{\text{2D}} &= K_{\text{recon}} \cdot E_{\text{recon}} \cdot \mathbf{l}_{\text{3D}}, \\ \mathbf{l}'_{\text{2D}} &= K_{\text{pigan}} \cdot E_{\text{pigan}} \cdot T_{\text{R2P}} \cdot \mathbf{l}_{\text{3D}}. \end{aligned} \tag{10}$$

Background Modeling. We model the background by setting the weight of the last sample point along each ray to be $w_N = 1 - \sum_{i=1}^{N-1} w_i$, where $w_i = T_i \cdot (1 - \exp(-\sigma_i \delta_i))$ is weight of the rest of the sample points along the ray, including both the mesh-guided samples and the volume samples.

Hyper-Parameters and Volume Sampling. Tab. 3 summarizes all hyper-parameters. We sample 12 coarse points evenly in the volume to and obtain $N_{\text{vol}} = 12$ fine points as introduced in [31]. Only $N_{\text{vol}} = 12$ fine points and the $N_{\text{surf}} = 12$ points sampled around the 3DMM input mesh are used for rendering and optimization with gradient backpropagation. The final model is trained for 72 hours on 8 GeForce GTX TITAN X GPUs.

To determine loss weights λs, we start from the original Pi-GAN and add the proposed components one-by-one, as in Tab. 2 in the main paper. For each component, we first initialize the hyperparameter lambda as 1, and empirically find a reasonable value before taking a fine-grained search.

B Additional Quantitative Results

B.1 Further Comparison with DiscoFaceGAN

In Tab. 1 in the main paper, we have compared with DiscoFaceGAN [8] in terms of Disentangle Scores. To directly compare the 3DMM conditioning methods proposed in our paper with those in the DiscoFaceGAN [8], we introduce the conditioning methods of DiscoFaceGAN, *i.e.* imitative loss, and contrastive loss, into Pi-GAN to build the "Pi-GAN + DiscoFaceGAN" model.

Comparison results are shown in the Tab. 4. Row 1 shows the Disentangle Scores of the officially released DiscoFaceGAN model concerning shapes, expressions, and poses. Row 2 corresponds to the constructed "Pi-GAN + DiscoFaceGAN" method. Comparing Row 1 and 2, we find the Pi-GAN backbone enhances the disentanglement of head pose and face shape while harming the expression disentanglement. Row 3 indicates the performance of our method, which outperforms the "Pi-GAN + DiscoFaceGAN" model in all metrics by a large margin, demonstrating the efficiency of the proposed conditioning method against the DiscoFaceGAN [8].

Parameter	Value/Range
Optimizer	Adam
Generator learning rate	6×10^{-5}
Discriminator learning rate	2×10^{-4}
Number of iterations	$80,000$
Batch size	64
N_{surf}	12
N_{vol}	12
training image size	64
Loss weight λ_{gan}	1
Loss weight λ_{recon}	4
Loss weight λ_{d}	10000
Loss weight λ_{ldmk}	10
Loss weight λ_{warp}	10
Loss weight $\lambda_{\text{smooth}}^{\text{norm}}$	1000
Loss weight $\lambda_{\text{smooth}}^{\text{depth}}$	500
$\mathbf{z}$	$\mathcal{N}^{411}(0,1)$
Ray length	$(0.88, 1.12)$
Yaw	$\mathcal{N}(0, 17.19°)$
Pitch	$\mathcal{N}(0, 8.88°)$
Field of view (FOV)	$12°$

Table 3: Training details and hyper-parameter settings.

B.2 User Study

We conduct a user study to add to the comparison. We follow the experiment setting of Tab. 1 in the main paper. Specifically, we provide the control results of five methods, and ask a total of 21 users to rank the results according to their image quality and the controlling effects.

Tab. 5 summarizes the results, where the "Average Ranking / Average Score / Ranking 1st Ratio" are reported for each method with respect to concerning factors, *i.e*. identity, expression, pose. Our model achieves the highest average ranking and scores for all aspects, indicating our model produces more perceptually compelling results and achieves better 3D controllability than other counterparts.

C Additional Qualitative Results

C.1 Additional Results with Pose Variations

We present large pose results for GAN-Control [46] in Fig. 6, DiscoFaceGAN [8] in Fig. 7, HeadNeRF [18] in Fig. 8 and pi-GAN [5] in Fig. 9. Previous methods fail to generate plausible face images when the camera pose gets larger (*e.g*. $> 60°$), including NeRF-based methods (HeadNeRF and pi-GAN). Nevertheless, as shown in Fig. 10, our method produces plausible *3D consistent* face images even in extremely large poses. Note that, to get rid of the view-dependent effect, we follow [14, 18] and remove the dependence of the radiance colors on the viewing direction by setting the view direction to a constant $(0, 0, -1)$ when evaluating the radiance colors of the sample points.

C.2 Additional Results with Expression Variations

We present more results on the expression control, comparing our method against two state-of-the-art controllable face synthesis methods, one attribute-guided GAN-Control [46], and the other 3DMM-guided DiscoFaceGAN [8]. Figs. 11 to 13 show the generated faces using GAN-Control, DiscoFaceGAN and our method respectively. For each figure, the first column shows a reference image, columns 2 to 5 show images generated with mild expressions, and columns 6 to 9 show images generated with wilder expressions. Each row corresponds to the same person, and each column corresponds to the same expression.

Index	Loss	CD ↓	LD ↓	LC ↑	DS_{s} ↑	DS_{e} ↑	DS_{p} ↑
1	Pi-GAN	1.09	5.04	2.04	2.13	2.54	7.16
2	DiscoFaceGAN [8]	-	-	-	5.97	15.70	5.23
3	Pi-GAN + DiscoFaceGAN	1.32	2.40	32.55	6.16	7.63	12.13
4	Ours	**0.27**	**1.26**	**92.88**	**23.24**	**29.13**	**23.45**

Table 4: Comparison with "Pi-GAN + DiscoFaceGAN".

	DiscoFaceGAN [8]	GAN-Control [46]	Pi-GAN [5] + $\mathcal{L}_{\text{recon}}$	HeadNeRF [18]	cGOF (Ours)
Id	3.1 / 59.1 / 9.52%	2.6 / 68.6 / 0.00%	3.3 / 53.3 / 9.52%	4.7 / 26.7 / 0.0%	**1.4 / 92.4 / 81.0%**
Exp	2.3 / 74.3 / 19.1%	3.8 / 43.8 / 9.52%	3.2 / 56.2 / 0.00%	4.3 / 34.3 / 0.0%	**1.4 / 91.4 / 71.4%**
Pose	2.6 / 68.6 / 14.3%	3.7 / 45.7 / 0.00%	3.8 / 43.8 / 4.76%	3.7 / 46.7 / 0.0%	**1.2 / 95.2 / 81.0%**
IQ	2.1 / 78.1 / 9.52%	2.9 / 62.9 / 4.76%	3.9 / 42.9 / 0.00%	5.0 / 20.0 / 0.0%	**1.2 / 96.2 / 85.7%**

Table 5: User Study on the Disentangle Performance and Image Quality. Each cell contains "Average Ranking / Average Score / Ranking 1st Ratio". Id: Identity, Exp: Expression, IQ: Image Quality.

In Fig. 11, we can see that GAN-Control fails to preserve the identity as well as other factors of the face image (*e.g*. background) when changing only the facial expression code. A few examples are highlighted in red. Moreover, we observe that with the original range of the expression parameters, the model results in only a small variation of expressions, whereas with increased perturbations, it leads to much more significant shape and identity inconsistencies. We also notice the "smiling" attribute tends to strongly correlate with the "female" and "long-hair" attributes.

In Fig. 12, we can see that the DiscoFaceGAN fails to impose consistent expression control over the faces. The blue boxes highlight a few examples, where the same expression code produces different expressions in different faces. Moreover, the images generated with wild expressions may appear unnatural, such as 'exp 5' in Fig. 12.

In Fig. 13, we show that our model generates compelling photo-realistic face images with highly consistent, precise expression control. In each row, only expressions change while other properties remain unchanged, such as identity (shape and texture), hair, and background. In each column, all instances follow the same expression.

In addition, in Fig. 14, we present examples of images generated by our model with *out-of-distribution* expressions, such as frowning, pouting, curling lips, smirking *etc*. Despite images with such expressions hardly existing in the training data, our model is still able to generate highly plausible images.

Figure 6: Images generated by DiscoFaceGAN [8] with different *poses*.

Figure 7: Images generated by GAN-Control [46] with different *poses*.

Figure 8: Images generated by HeadNeRF [18] with different *poses*.

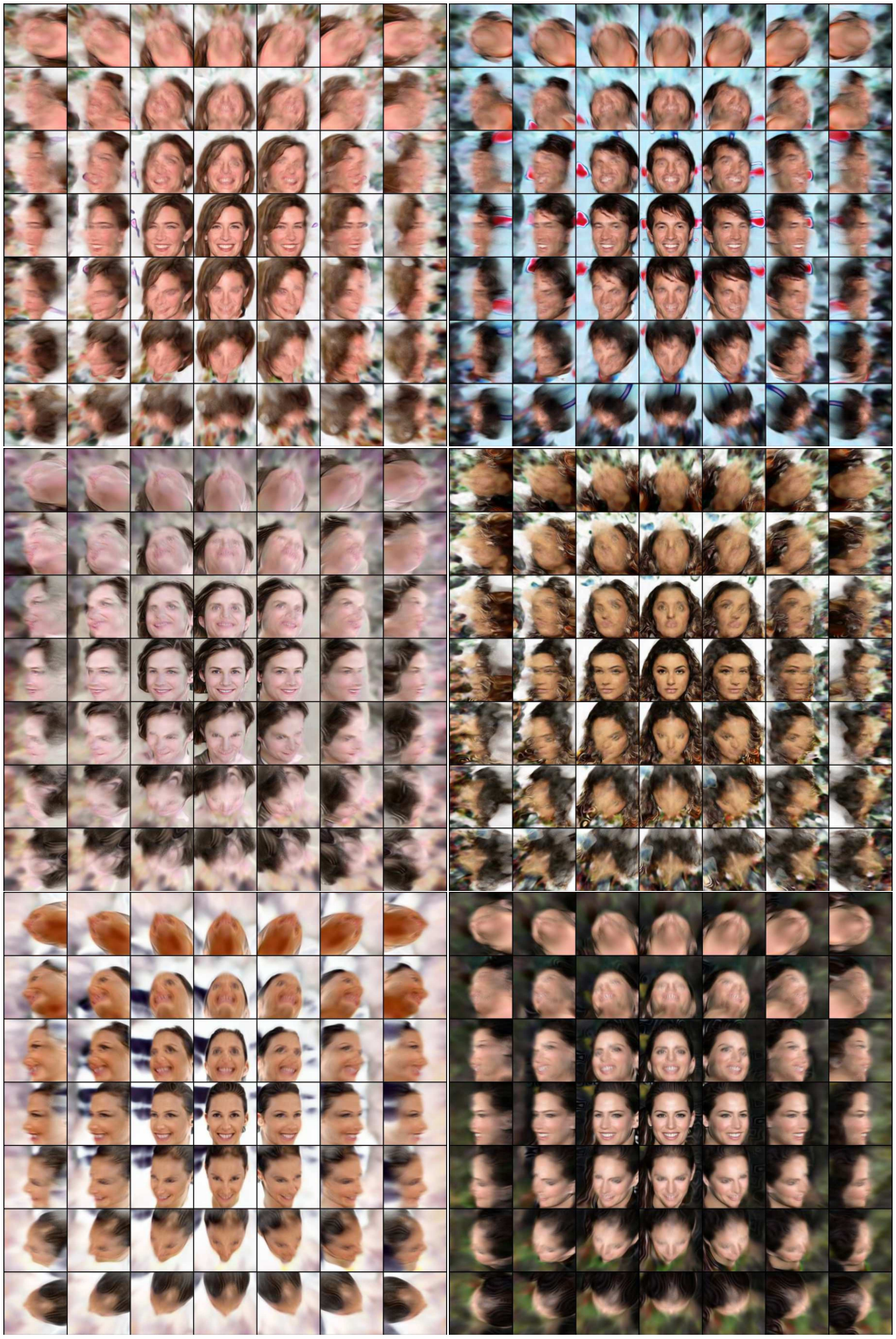


Figure 9: Images generated by the original Pi-GAN [5] with different *poses*.

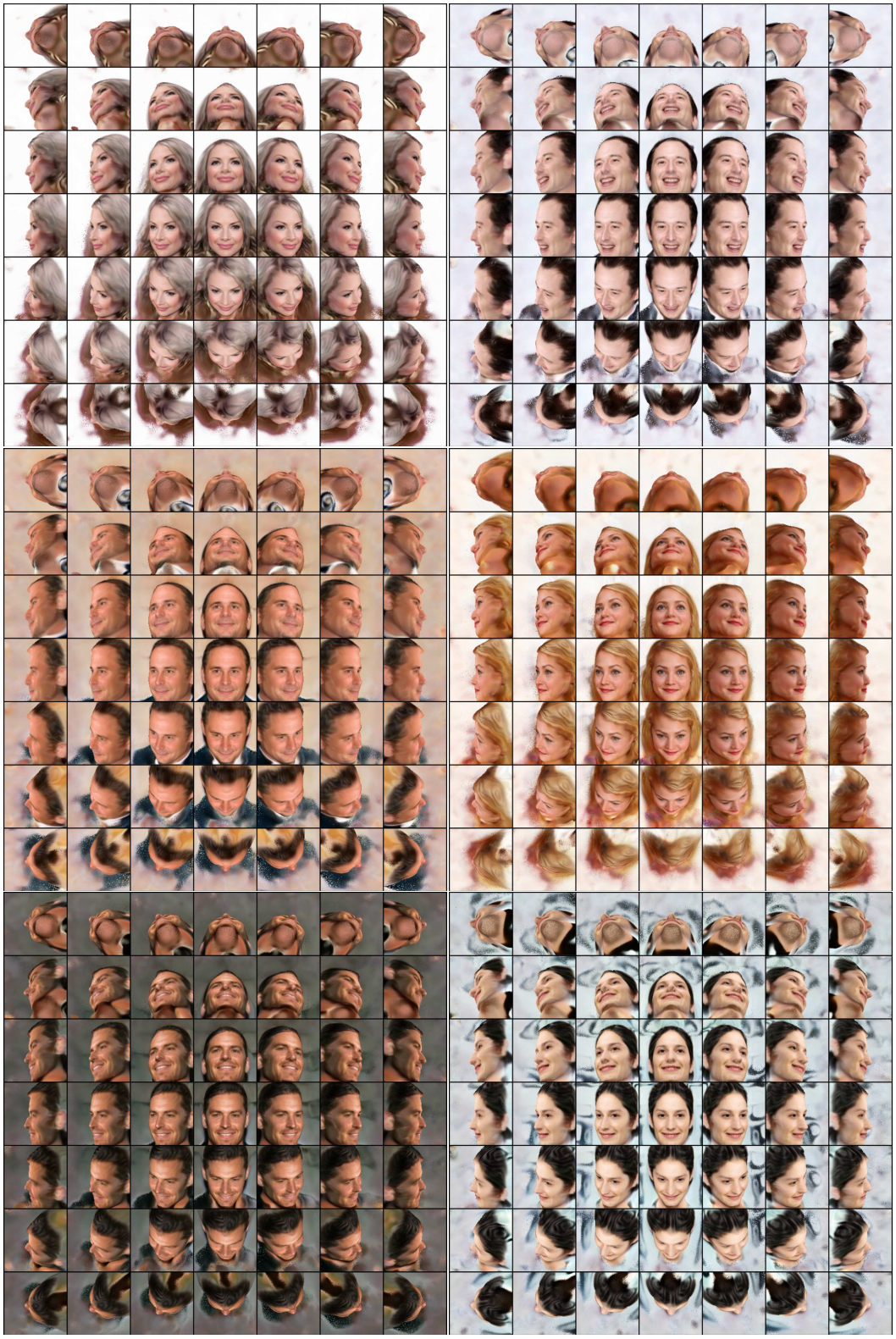


Figure 10: Images generated by our proposed cGOF with different *poses*.

Figure 11: Images generated by GAN-Control [46] with different *expressions*. Each row is generated with the same parameters except the expression code, and each column shares the same expression code. Red boxes highlight examples where GAN-Control produces severe inconsistencies in the identity and background when only the expression is supposed to change. Moreover, we notice the "smiling" attribute tends to strongly correlate with the "female" and "long-hair" attributes.

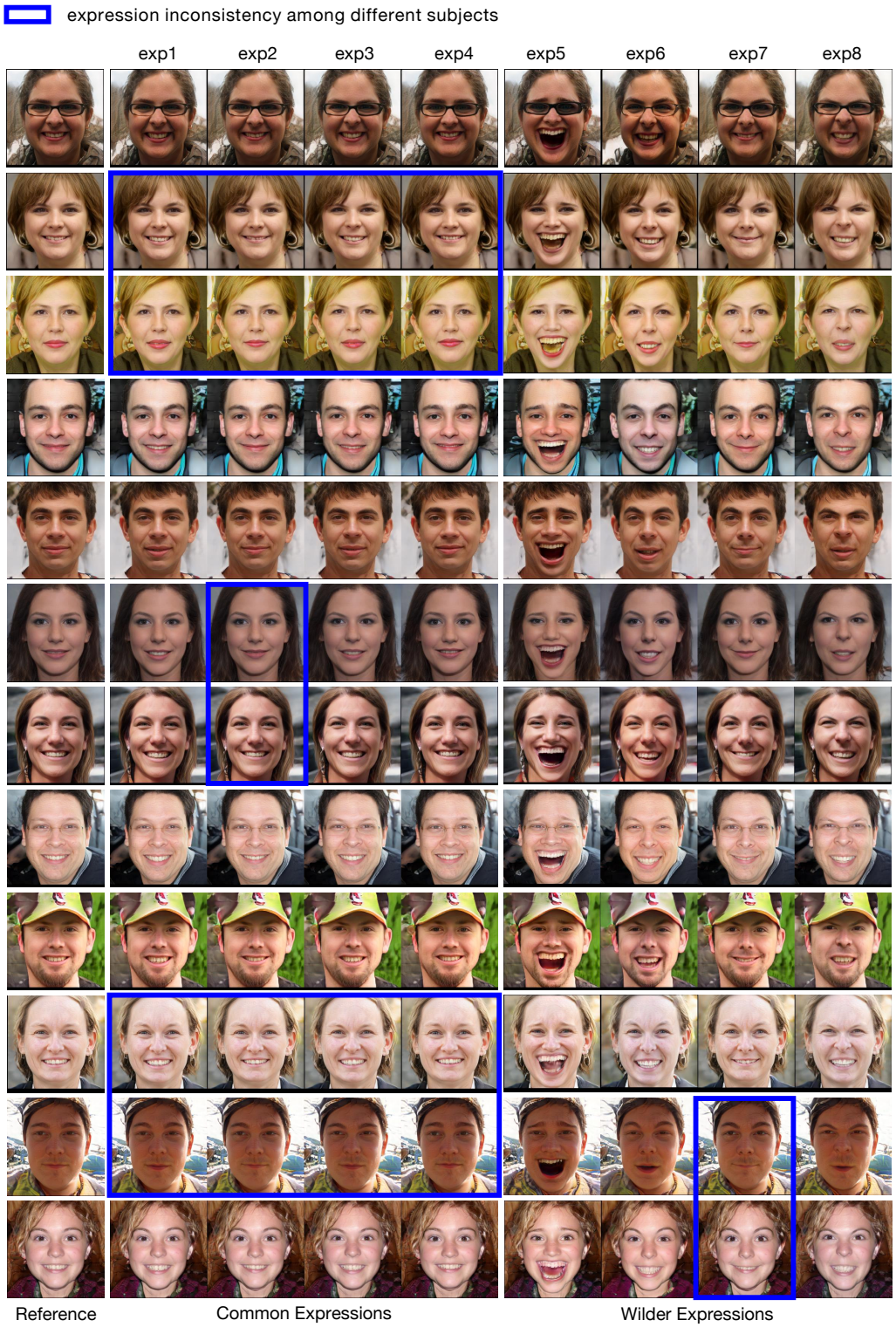


Figure 12: Images generated by DiscoFaceGAN [8] with different *expressions*. Each row is generated with the same parameters except the expression code, and each column shares the same expression code. Blue boxes highlight some examples where DiscoFaceGAN fails to produce consistent expressions among different instances.

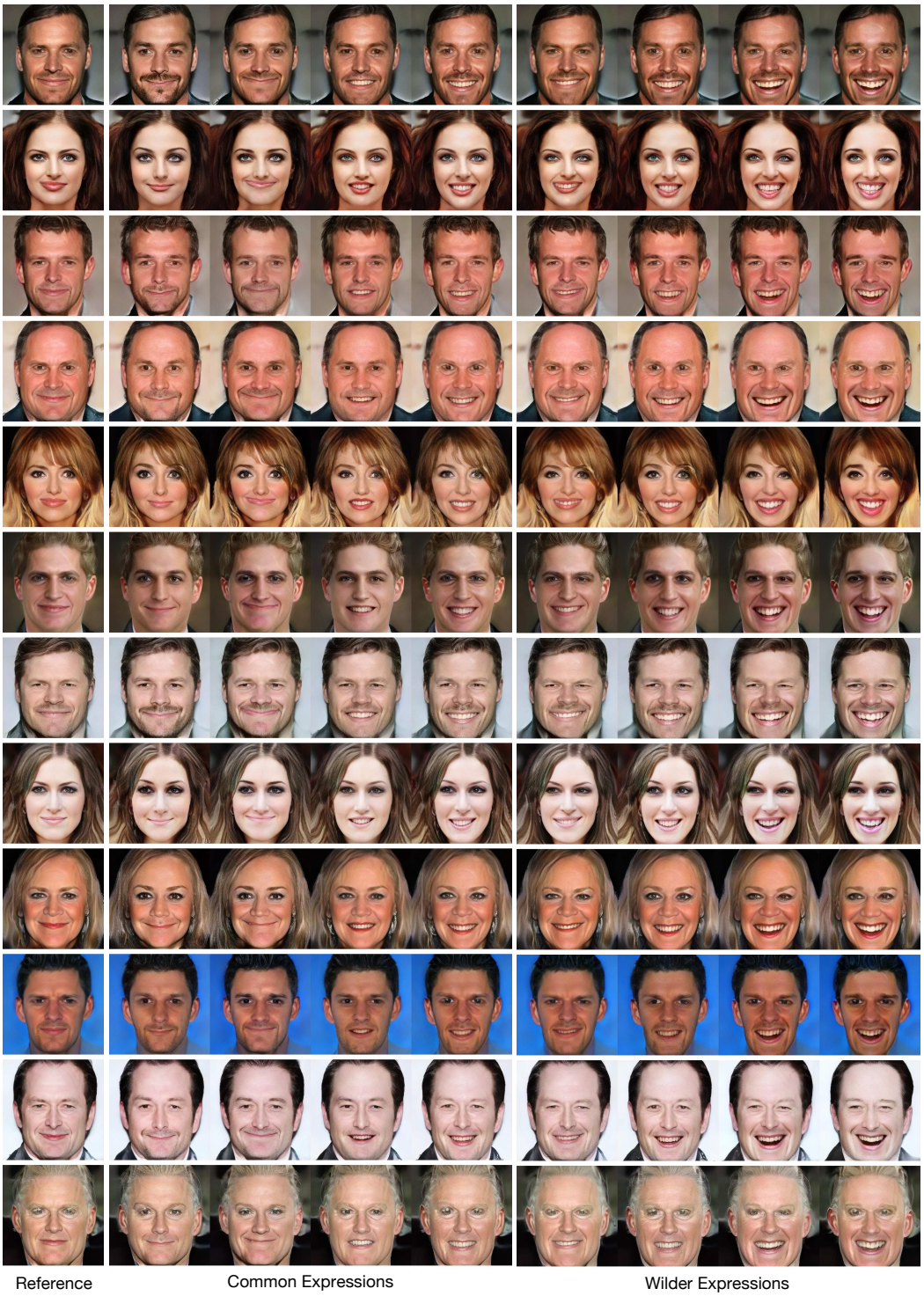


Figure 13: Images generated by our proposed cGOF with different *expressions*. Our model generates photo-realistic face images with highly consistent, precise expression control.

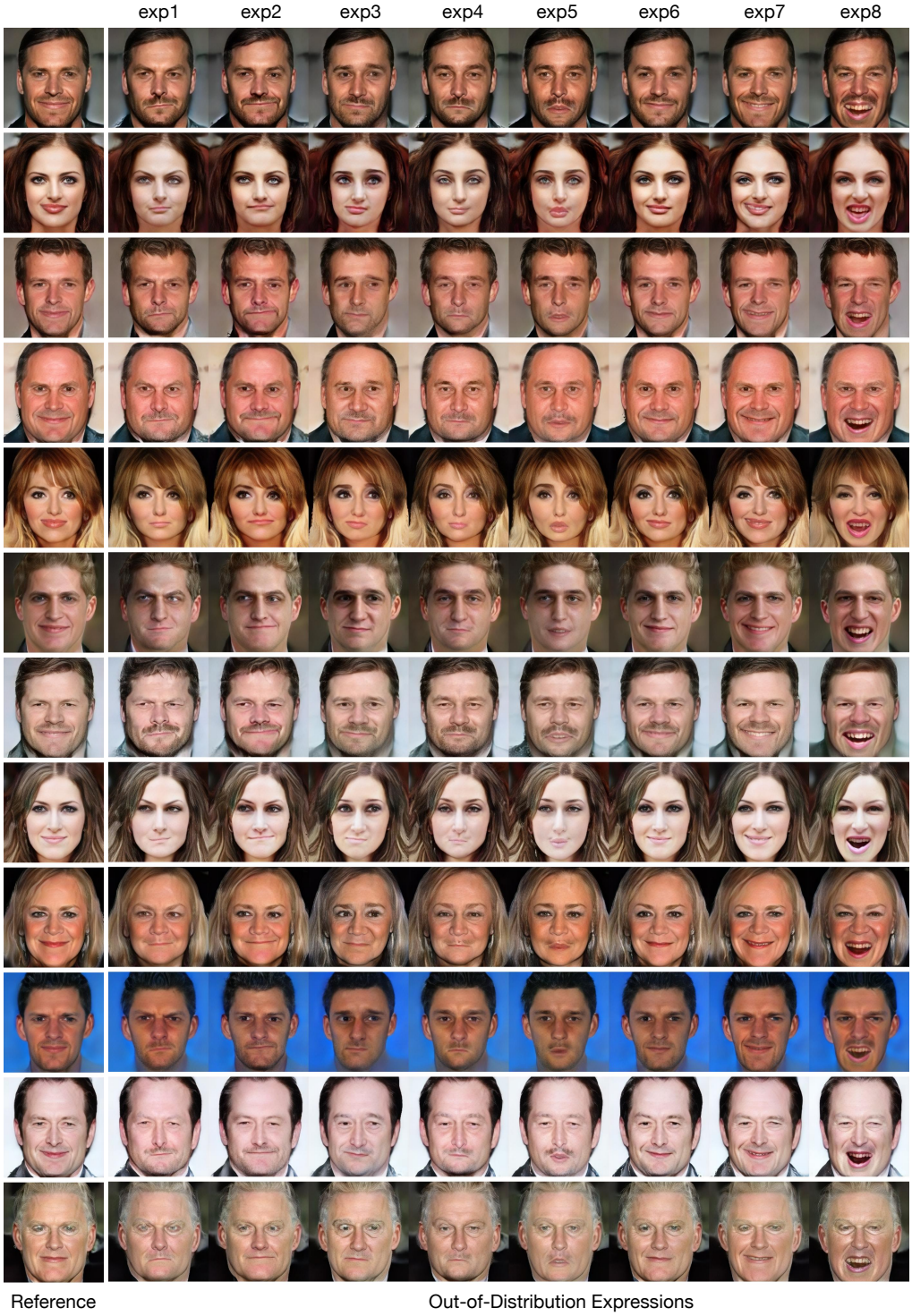


Figure 14: Images generated by our proposed cGOF with *out-of-distribution expressions*. Our method is capable of synthesizing unseen expressions like raising eyebrow, pouting, curling lips, smirking *etc*.