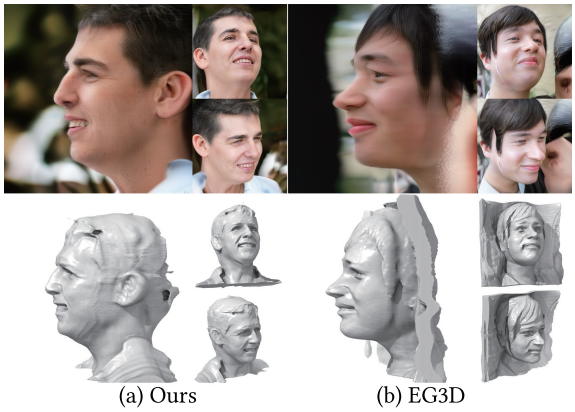
LPFF: A Portrait Dataset for Face Generators Across Large Poses

Yiqian Wu[1] Jing Zhang[1] Hongbo Fu[2] Xiaogang Jin[1*]
[1]State Key Lab of CAD&CG, Zhejiang University
[2]City University of Hong Kong
onethousand@zju.edu.cn, jing_z99@163.com, hongbofu@cityu.edu.hk, jin@cad.zju.edu.cn

Abstract

Existing face generators exhibit exceptional performance on faces in small to medium poses (with respect to frontal faces) but struggle to produce realistic results for large poses. The distorted rendering results on large poses in 3D-aware generators further show that the generated 3D face shapes are far from the distribution of 3D faces in reality. We find that the above issues are caused by the training dataset's pose imbalance. To this end, we present LPFF, a large-pose Flickr face dataset comprised of 19,590 high-quality real large-pose portrait images. We utilize our dataset to train a 2D face generator that can process large-pose face images, as well as a 3D-aware generator that can generate realistic human face geometry. To better validate our pose-conditional 3D-aware generators, we develop a new FID measure to evaluate the 3D-level performance. Through this novel FID measure and other experiments, we show that LPFF can help 2D face generators extend their latent space and better manipulate the large-pose data, and help 3D-aware face generators achieve better view consistency and more realistic 3D reconstruction results.



(a) Ours (b) EG3D

Figure 1: Image and shape samples generated by EG3D models [5] trained with the same training strategy but using different datasets (our new dataset *LPFF* and *FFHQ* for (a) and *FFHQ* for (b)). The generators are conditioned by the average camera parameters. Shapes are iso-surfaces extracted from the corresponding density fields using marching cubes. Our dataset helps reduce distorted, "seam", "wall-mounted", and blurry artifacts exhibited in (b).

1. Introduction

Since the first introduction by Goodfellow in 2014, generative adversarial networks (GANs) [11] have significantly advanced the performance of 2D high-resolution image generation. GANs can accomplish a variety of downstream image editing tasks, particularly face modification [2, 16, 42, 43], thanks to the excellent image quality and semantic features in its latent space. Recently, plenty of 3D-aware generators [12, 30, 5, 52, 35, 9, 41, 40] have been proposed to learn 3D-consistent face portrait generation from 2D image datasets. 3D-aware generators can describe and represent geometry in their latent space while rendering objects from different camera perspectives using volumetric rendering. Researchers carefully designed generator architectures and training strategies to accelerate training, reduce memory overheads, and increase rendering resolution.

Both the existing 2D and 3D approaches, however, are unable to process large-pose face data. Regarding 2D face generators, those large-pose data are actually outside of their latent space, which prevents them from generating reasonable large-pose data, thus causing at least two problems. First, as shown in Fig. 2 (left), moving the latent code along the yaw pose editing direction will cause it to reach the edge of the latent space before faces become profile. Second, as shown by the results of image inversion in Fig. 2 (right), it is challenging to project large-pose images to the latent space, let alone perform semantic modification on them. One of the goals of 3D-aware generators is to model realistic human face geometry, but existing 3D-aware generators trained on 2D image datasets still have difficulty producing realistic geometry. This issue is more serious when render-

*Corresponding author.

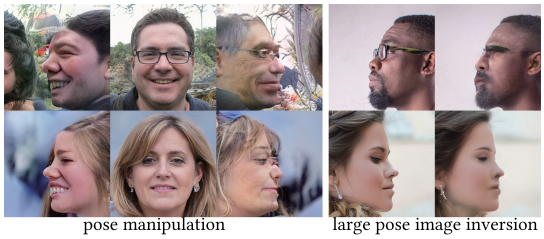


Figure 2: StyleGAN2 [22]'s large-pose performance when trained on *FFHQ*. InterfaceGAN [38] is used to edit the yaw angle of randomly sampled latent codes. We use optimization-based GAN inversion to obtain the latent codes of target large-pose real images.

ing the results at extreme poses. As shown in Fig. 3, faces synthesized by those methods have noticeable artifacts, including distortion, blurring, and stratification. In Fig. 1 (b), EG3D shows a "wall-mounted" and distorted 3D representation without ears. All these indicate that the generated face shapes are not realistic enough.

The above issues in the face generators are mainly caused by the unbalanced camera pose distribution of the narrow-range training dataset. **F**lickr-**F**aces-**HQ** Dataset (**FFHQ**) is a popular high-quality face dataset used to train those face generators, but it mainly contains images limited to small to medium poses. As a result, 2D and 3D-aware generators cannot learn a correct large-pose face distribution without sufficient large-pose data. To avoid artifacts under large poses, downstream applications [42, 28, 24, 46, 16, 17, 1, 49] based on those face generators typically sample small poses, which limits their application scenarios.

It is difficult to get a pose-balanced dataset. First, large-pose faces are nearly impossible to detect using Dlib [23], a popular face detector, and the one used to crop *FFHQ*. Second, simply replicating extremely limited large-pose data to balance the pose distribution is insufficient to help extend the camera distribution. As a result, it is critical to collect a large number of large-pose, in-the-wild, and high-resolution face images, which are lacking in existing datasets.

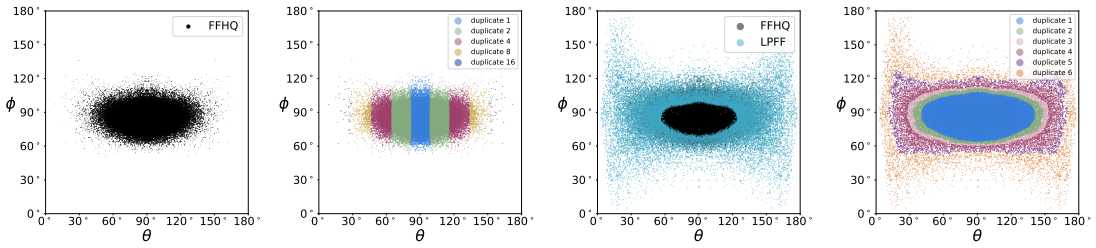
In this paper, we propose a novel high-quality face dataset containing **19,590** real large-pose face images, named **L**arge-**P**ose-**F**lickr-**F**aces Dataset (**LPFF**), as a supplement to *FFHQ*, in order to extend the camera pose distribution of *FFHQ* and train 2D and 3D-aware face generators that are free of the aforementioned problems. Given the difficulty of large-pose face detection and the imbalanced distribution of camera poses in real-life photographs, we design a face detection and alignment pipeline that is better suited to large-pose images. Our method can also gather large amounts of large-pose data based on pose density. We retrain StyleGAN2-ada [19] to demonstrate how our dataset can assist 2D face generators in generating and editing large-pose faces. We retrain EG3D [5] as an example to demonstrate how our dataset can aid 3D face generators in understanding realistic face geometry and appearance across a wide range of camera poses. In order to better evaluate the 3D-level performance of EG3D models trained on different datasets, we propose a new FID measure for pose-conditional 3D-aware generators. Extensive experiments show that our dataset leads to realistic large-pose face generation and manipulation in the 2D generator. Furthermore, our dataset results in more realistic face geometry generation in the 3D-aware generator.

Figure 3: 3D-aware generators trained on *FFHQ* (StyleNeRF [12], StyleSDF [30], EG3D [5], and IDE-3D [42]) achieve excellent image synthesis performance on faces in small to medium poses (Top), but exhibit obvious artifacts at steep angles (Bottom).

Our paper makes the following major contributions: 1) A novel data processing and filtering method that can collect large-pose face data from the Flickr website according to camera pose distribution, leading to a novel face dataset that contains 19,590 high-quality real large-pose face images. 2) A retrained 2D face generator that can process large-pose face images. 3) A retrained 3D-aware generator that can generate realistic human face geometry. 4) A new FID measure for pose-conditional 3D-aware generators.

2. Related Work

2D Face Generators. Since Goodfellow first proposed the generative adversarial networks (GANs) in 2014 [11], many GANs model designs [32, 13, 3, 18] have been developed to produce more impressive performance on realistic image synthesis. Among these GANs models, StyleGAN [21, 22, 19, 20] is regarded as the most cutting-edge generator of high-quality images. For face portrait images, StyleGAN provides not only realistic image generation but also implicit semantic features in latent space, which are beneficial for many downstream computer vision applications [2, 7, 31]. However, the face StyleGAN is trained on a pose-imbalanced face dataset, *FFHQ*. StyleGAN inherits the pose bias from *FFHQ*, resulting in artifacts and distortions when projecting and editing large-pose portraits. This issue is particularly noticeable in real-world applications, where photographs are not always forward-facing.



(a) (b) (c) (d)

Figure 4: (a) *FFHQ*. (b) *FFHQ-rebal*. (c) *FFHQ+LPFF*. (d) *FFHQ+LPFF-rebal*. The term "duplicate" refers to the number of repetitions in data resampling.

3D-aware Face Generators. 2D generators have been expanded to support 3D multi-view rendering. Early methods combined voxel rendering [29, 14, 53] or NeRF rendering [34, 12, 6] with generators to support view-consistent image synthesis. However, those methods have a high cost of calculation, which limits the resolution of the output. Later, a number of studies suggested an additional super-resolution network to enhance image quality without adding too much computing load [5, 44, 30, 47]. Researchers also suggested an effective optimization strategy [41] to directly output high-resolution results without any super-resolution module. These techniques not only produce view-consistent results, but also learn, represent, and generate face geometry in a generative manner. Several methods [42, 16] achieve semantic attribute editing and geometry-appearance disentanglement in view-consistent image synthesis through the integration of semantic masks into 3D-aware generators. However, for human face training, they rely heavily on 2D image collections (*FFHQ*). Because of the pose imbalance in the training dataset, incorrect facial geometry may arise in the final results, which negatively impacts the performance of downstream applications [42, 28, 24, 46, 16, 17, 1, 49]. Researchers also try to eliminate the dependency for 3D pose priors [39] or resample the dataset to increase the density of extremely limited large-pose data [5]. However, both of them cannot address the root causes.

Face Image Datasets. Numerous studies have noted the pose imbalance in current face image datasets. 300W-LP [54] is a dataset consisting of 61,225 images across large poses, but all the images are artificially synthesized by face profiling. AFLW [25] contains 21,080 face images with large-pose variations, LS3D-W [4] contains ∼230,000 images from a combination of 300-W test set [33], 300-VW [37], Menpo [50], and AFLW2000-3D [54]. But most images in AFLW and LS3D-W are at low resolution. There are several 3D face datasets [48, 45, 27] that contain high-quality multi-view face images mainly for 3D face reconstruction. However, these datasets have limited variety [48], or are low-resolution [25, 4], or are synthesized artificially [54]. In contrast, our dataset consists of high-resolution real images collected from in-the-wild photographs.

3. Data Preparation

In this section, we will introduce how to build our large-pose face dataset. First, we describe the process for extracting data density from *FFHQ* (Sec. 3.1). Then, we introduce a novel data processing pipeline that can produce more reasonable realigned results (Sec. 3.2). In order to filter large-pose face data from the Flickr images according to camera distribution, we propose to employ the pose density function to collect only large-pose data (Sec. 3.3). Finally, we introduce a novel rebalancing strategy (Sec. 3.4).

3.1. Camera Distribution

EG3D uses a face reconstruction model [10], denoted as $\mathcal{F}$ in this paper, to extract camera parameters. All cameras are assumed to be positioned on a spherical surface with a radius $r = 2.7$, and the camera intrinsics are fixed. In this paper, we only consider the camera location and ignore the roll angle of the camera to compute the camera distribution (detailed in the supplementary file). We convert the coordinates of each camera in *FFHQ* from Cartesian coordinates to spherical coordinates and get their θ and ϕ (see Fig. 4 (a)). Notice that the face with $\theta = 90°$ and $\phi = 90°$ is frontal.

3.2. Data Processing

Given the difficulty of large-pose face detection and the imbalanced distribution of camera poses in real-life photographs, we propose a novel mechanism to collect, process, and filter large-pose data. We first collect **155,720** raw portrait images from Flickr[1] (with permission to copy, modify, distribute, and perform). Then we remove all the raw images that already appeared in *FFHQ*.

[1] https://www.flickr.com

Our pipeline is based on that of EG3D, and we respectively align each raw image according to the image align function in EG3D and StyleGAN. In EG3D, the authors first predict the 68 face landmarks of a raw image by Dlib, and then get a realigned image by using the eyes and mouth positions to determine a square crop window for cropping and rotating the raw image. The realigned image is denoted as $X_{realigned}$ with the eyes at the horizontal level and the face at the center of the image. Then MTCNN [51] is used to get the positions of the eyes, the nose, and the corners of the mouth of $X_{realigned}$, and the 5 feature points are then fed into $\mathcal{F}$ to predict camera parameters. Finally, these positions are used to crop $X_{realigned}$, resulting in the final image.

In our pipeline, we first use Dlib to get 68 landmarks for each of the 155,720 raw portrait images, and for those images that resist face detection, we additionally apply face alignment [4] (SFD face detector) to predict landmarks. The face alignment detector achieves better performance on large-pose face detection than Dlib. Joining the two landmark predictors can help us detect as many large-pose faces as possible. Then the predicted landmarks are used to get the realigned image $X_{realigned}$. In this step, we get **506,262** $X_{realigned}$.

We find that the MTCNN sometimes cannot predict landmarks for large-pose faces. So instead of using MTCNN, we directly aggregate the 68 landmarks to get the 5 feature points of the eyes, mouth, and nose.

After that, we use $\mathcal{F}$ to predict camera parameters. Then we filter large-pose face data from **506,262** $X_{realigned}$ (detailed in Sec. 3.3), getting **208,543** large pose $X_{realigned}$. We automatically filter out low-resolution images and manually examine the rendering results of the reconstructed face models, removing any failed 3D reconstructions (which indicate incorrectly estimated camera parameters), as well as blurry or noisy images. Finally, we get **19,590** high-quality large-pose face images with correctly estimated camera parameters.

When cropping the final image, we find that some of the 5 feature points (especially when there is a face with eyeglasses) are not accurate enough to crop $X_{realigned}$ properly, but after manual selection, the landmarks that $\mathcal{F}$ produces are more aligned with the input faces. So we use the landmarks of the reconstructed face to crop $X_{realigned}$ according to EG3D and StyleGAN functions and obtain final images. Please refer to the supplement file for an illustration of the image processing pipeline.

3.3. Large-Pose Data Selection

To collect only images with "low density" (at large poses), we propose using the density function of FFHQ to filter large pose faces. Inspired by [26], we estimate the density of the *FFHQ* camera (θ, ϕ) tuples using Gaussian kernel density estimation and Scott's rule [36] as a bandwidth selection strategy. After obtaining ρ_{ffhq}, where $density = \rho_{ffhq}(\theta, \phi)$ is the density of the camera at (θ, ϕ), we use ρ_{ffhq} to compute the density of **506,262** $X_{realigned}$, and filter the images with a density less than 0.4 ($density = \rho_{ffhq}(\theta, \phi) < 0.4$).

3.4. Data Rebalance

After image processing, large pose filtering, and carefully manual selecting, we get **19,590** large-pose face images as our *LPFF* dataset. We use the *LPFF* dataset as a supplement to *FFHQ*. That is, we combine *LPFF* with *FFHQ*, named *FFHQ+LPFF*. The datasets are augmented by a horizontal flip. In Fig. 4, we show the camera distribution for both *FFHQ+LPFF* and *FFHQ*.

To improve our models' performance on large-pose rendering quality and image inversion, we propose using a resampling strategy to further rebalance our *FFHQ+LPFF* dataset (refer to Sec. 5 for evaluation). In EG3D, in order to increase the sampling probability of the low-density data, the authors rebalanced the *FFHQ* dataset by splitting it into 9 uniform-sized bins across the yaw range and duplicating the images according to the bins (as shown in Fig. 4 (b)). We denoted the rebalanced *FFHQ* dataset as *FFHQ-rebal*.

Inspired by EG3D, we also rebalance *FFHQ+LPFF* to help the model focus more on large-pose data. Instead of simply splitting the dataset according to yaw angles, we split *FFHQ+LPFF* according to the data densities (Fig. 4 (d)). Similar to Sec. 3.1, we first compute the pose density function of *FFHQ+LPFF* (denoted as $density = \rho_{ffhq+lpff}(\theta, \phi)$), then duplicate our dataset as:

$$\begin{cases} N = \min(\max(\text{round}(\frac{\alpha}{density}), 1), 4), density \geq 0.03 \\ N = 5, density \in [0.02, 0.03) \\ N = 6, density \in [0, 0.02) \end{cases} \tag{1}$$

where α is a hyper-parameter (we empirically set $\alpha = 0.24$ in our experiments), and N denotes the number of repetitions. The rebalanced *FFHQ+LPFF* is denoted as *FFHQ+LPFF-rebal*.

4. Training Details

In this section, we will retrain 2D and 3D-aware face generators using our dataset. Regarding the 2D generator (Sec. 4.1), we retrain StyleGAN2-ada using our dataset before fine-tuning the model using the rebalanced dataset. As for the 3D-aware generator (Sec. 4.2), we first use our dataset to retrain EG3D, and then use the rebalanced dataset to fine-tune the model. In order to improve image synthesis performance during testing, we further fine-tune the model by setting the camera parameters input to the generator as the average camera.

4.1. StyleGAN

Retrain. In the StyleGAN training, we use the StyleGAN2-ada architecture as our baseline, and train it on *FFHQ+LPFF* from scratch. We use the training parameters defined by *stylegan2* config in StyleGAN2-ada. We denote the StyleGAN2-ada model trained on *FFHQ* as S^{FFHQ}_{var1}, and the model trained on *FFHQ+LPFF* as S^{Ours}_{var1}. Our training time is $\sim$5 days on 8 Tesla V100 GPUs.

Rebalanced dataset fine-tuning. We utilize the rebalanced dataset, *FFHQ+LPFF-rebal*, to fine-tune S^{Ours}_{var1}, and denote the rebalanced model as S^{Ours}_{var2}. All training parameters are identical to those of S^{Ours}_{var1}. Our fine-tuning time is $\sim$18 hours on 8 Tesla V100 GPUs.

4.2. EG3D

The mapping network, volume rendering module, and dual discriminator in EG3D [5] are all camera pose-dependent. We divided the EG3D model into three modules: Generator G, Renderer R, and Discriminator D, please refer to the supplement file for an illustration of the three modules. The attribute correlations between pose and other semantic attributes in the dataset are faithfully modeled by using the camera parameters fed into G. R and D are always fed with the same camera specifications. The camera parameters help D ensure multi-view-consistent super resolution and direct R in how to render the final images from various camera views.

In this paper, we define two types of camera parameters that are inputted into the whole model as:

$$c = [c_g, c_r], \tag{2}$$

where c_g stands for the camera parameters fed into G, and c_r stands for the camera parameters fed into R and D. c_g will influence the face geometry and appearance and should be fixed in testing. The authors of EG3D discover that maintaining $c_g = c_r$ throughout training can result in a GAN that generates 2D billboards. To solve this problem, they apply a swapping strategy that randomly swaps c_g with another random pose in that dataset with β probability, where β is a hyper-parameter.

Retrain. We use *FFHQ+LPFF* to train EG3D from scratch. All the training parameters are identical to those of EG3D, where β is linearly decayed from 100%to 50% over the first 1M images, and then fixed as 50% in the remaining training. We denote the EG3D trained on *FFHQ* as E^{FFHQ}_{var1} (the original EG3D), denote the EG3D trained on *FFHQ+LPFF* as E^{Ours}_{var1}. Our training time is $\sim$6.5 days on 8 Tesla V100 GPUs.

Rebalanced dataset fine-tuning. In EG3D, the authors use the rebalanced dataset *FFHQ-rebal* to fine-tune E^{FFHQ}_{var1}, leading to a more balanced model. We denote the fine-tuned model as E^{FFHQ}_{var2}. For a fair comparison, we also use the same fine-tuning strategy as EG3D to fine-tune our model E^{Ours}_{var1} on our rebalanced dataset *FFHQ+LPFF-rebal*. β is fixed as 50% in training, and other training parameters are identical to those of EG3D. We denote E^{Ours}_{var1} fine-tuned on *FFHQ+LPFF-rebal* as E^{Ours}_{var2}. Our fine-tuning time is $\sim$1 day on 8 Tesla V100 GPUs.

Figure 5: Images produced by our S^{Ours}_{var1} model (Top) and S^{Ours}_{var2} model (Bottom). We apply truncation with $\psi = 0.7$.

5. Evaluation

To show that *LPFF* can help 2D and 3D-aware face generators generate realistic results across large poses, we will first evaluate the performance of 2D face generators (Sec. 5.1), and then demonstrate the performance of 3D-aware face generators (Sec. 5.2).

5.1. StyleGAN

Fig. 5 shows the uncurated samples of faces generated by the models trained on our dataset, with resolution 1024^2. Our models synthesize images that are of high quality and have large pose variance.

FID and perceptual path length. We trained the models using different datasets, so the latent space distributions are different in our experiments. Therefore, we do not compare the Fréchet Inception Distance (FID) [15] and perceptual path length (PPL) [21] between the models, since they are highly related to dataset distributions. Instead, we respectively measure the FID of S^{FFHQ}_{var1}, S^{Ours}_{var1} and S^{Ours}_{var2} on their training dataset. The FID of S^{FFHQ}_{var1} is 2.71 on *FFHQ*, the FID of S^{Ours}_{var1} is 3.407 on *FFHQ+LPFF*, and the FID of S^{Ours}_{var2} is 3.786 on *FFHQ+LPFF-rebal*. The comparable FIDs show that the StyleGAN2-ada model can achieve convergence on our datasets as it did on *FFHQ*. We use the PPL metric that is computed based on path endpoints in W latent space, without the central crop. The PPL of S^{FFHQ}_{var1} is 144.9, the PPL of S^{Ours}_{var1} is 147.6, and the PPL of S^{Ours}_{var2} is 173.0. The PPL of S^{Ours}_{var1} is comparable to the PPL of S^{FFHQ}_{var1}. The higher PPL of S^{Ours}_{var2} indicates that S^{Ours}_{var2} leads to more drastic image feature changes when performing interpolation in the latent space. We attribute this to the larger pose variance in S^{Ours}_{var2}'s latent space and

Figure 6: Pose manipulation comparison between S_{var1}^{FFHQ} (Top), S_{var1}^{Ours} (Middle), and S_{var2}^{Ours} (Bottom). The images highlighted by the blue box are generated from randomly sampled latent codes, and all the samples are linearly moved along the yaw editing direction with the same distance.

the *FFHQ+LPFF-rebal* dataset.

Pose manipulation. We compare the pose distribution of the latent spaces by displaying the results of linear yaw pose manipulation. For each model, we label randomly sampled latent codes according to the camera parameters of the corresponding synthesized images (yaw angles $>90°$ as positive and $\leq 90°$ as negative) and use InterfaceGAN [38] to compute the yaw editing direction. The pose editing results are then obtained by moving randomly sampled latent codes along the yaw editing direction, as shown in Fig. 6. Because the linear manipulation method is used without any semantic attribute disentanglement, the results of all models cannot preserve facial identity. As for S_{var1}^{FFHQ}, the "side face" results are far from a genuine human face, demonstrating that the latent codes have reached the edge of the latent space. With regard to S_{var1}^{Ours} and S_{var2}^{Ours}, our models produce reasonable and comparable large-pose portraits. The comparison shows that our models' latent spaces are more extensive and better able to represent large-pose data.

Large-pose data inversion and manipulation. To further show that our models can better represent large-pose data, we project large-pose portraits into the latent spaces of those models (see Fig. 7), and apply semantic editing to the obtained latent codes. We collect the testing images from Unsplash[2] and Pexels[3] (independent of both *FFHQ* and *LPFF*). We then employ 500-step latent code optimization in $W+$ latent space to minimize the distance between the synthesized image and the target image. To evaluate the editability of the projected latent codes, we use the attribute classifiers provided by StyleGAN [21] and employ InterfaceGAN to compute semantic boundaries for each model, and then use the boundaries to edit the projected latent codes. We also use the yaw editing directions to try to make the large pose data face forward. Please refer to the supplement for those semantic editing results. As shown in those projection and manipulation results, the models trained on our dataset have fewer artifacts and can better represent the large pose data in their latent spaces. What's more, S_{var2}^{Ours} outperforms S_{var1}^{Ours} because S_{var2}^{Ours} is trained on a more balanced dataset, which proves the effectiveness of our data rebalance strategy.

[2]https://unsplash.com

[3]https://www.pexels.com

Figure 7: Large-pose data projection comparison between S_{var1}^{FFHQ}, S_{var1}^{Ours}, and S_{var2}^{Ours}. The target images (the first row) are collected from Unsplash and Pexels websites.

5.2. EG3D

Fig. 8 provides the selected samples that are generated by the models trained on the *FFHQ* dataset and our dataset, with resolution 512^2. Even in large poses, our synthesized images and 3D geometry are high-quality.

FID. In EG3D, the generator is conditioned on a fixed camera pose (c_g) when rendering from a moving camera trajectory to prevent the scene from changing when the camera (c_r) moves during inference. However, EG3D's authors evaluated the FID of EG3D by conditioning the model on c_g and rendering results from $c_r = c_g$. This approach cannot demonstrate the performance of multi-view rendering during inference, since the generator always "sees" the true pose of the rendering camera in evaluation, but omits other poses. For a 3D-aware generator, we are more interested in how a face looks from various camera views (which can indicate the quality of face geometry to some extent). So a more reasonable way is to let c_r and c_g be independent of each other and sample them from the respective distributions that are of our interest. To achieve this, we propose a novel FID measure, which is based on three camera sampling strategies. First, we fix c_g as c_{avg} and then sample c_r from different datasets. Second, we respectively sample c_r and c_g from different datasets. Third, we sample c_g from different datasets and set $c_r = c_g$ (the one that was used in

Figure 8: Image-shape pairs produced by E^{FFHQ}_{var1},E^{FFHQ}_{var2}, E^{Ours}_{var1}, and E^{Ours}_{var2}. We apply truncation with $\psi = 0.8$.

model	$c_g = c_{avg}$ $c_r \sim$ FFHQ	$c_g \sim$ FFHQ $c_r \sim$FFHQ	$c_g \sim$ LPFF $c_r \sim$FFHQ
E^{FFHQ}_{var1}	0.771	0.768	0.760
E^{Ours}_{var1}	**0.804**	**0.792**	**0.778**
E^{FFHQ}_{var2}	0.770	0.769	0.766
E^{Ours}_{var2}	**0.789**	**0.784**	**0.771**

Table 1: Quantitative evaluation of facial identity consistency (↑).

model	$c_g = c_{avg}$ $c_r \sim$ FFHQ	$c_g \sim$ FFHQ $c_r \sim$FFHQ	$c_g \sim$ LPFF $c_r \sim$FFHQ
E^{FFHQ}_{var1}	0.134	0.133	0.159
E^{Ours}_{var1}	**0.119**	**0.124**	**0.134**
E^{FFHQ}_{var2}	0.135	0.130	0.142
E^{Ours}_{var2}	**0.117**	**0.122**	**0.131**

Table 2: Quantitative evaluation of geometry consistency (↓).

model	$c_g = c_{avg}$ $c_r \sim$ FFHQ	$c_g = c_{avg}$ $c_r \sim$ LPFF	$c_g \sim$ FFHQ $c_r \sim$FFHQ	$c_g \sim$ FFHQ $c_r \sim$LPFF	$c_g \sim$ LPFF $c_r \sim$FFHQ	$c_g \sim$ LPFF $c_r \sim$LPFF	$c_g \sim$ FFHQ $c_r = c_g$	$c_g \sim$ LPFF $c_r = c_g$
E^{FFHQ}_{var1}	**6.523**	23.598	**4.273**	22.318	23.698	36.641	**4.025**	23.301
E^{Ours}_{var1}	7.997	**20.896**	6.623	**19.738**	**21.300**	**22.074**	6.093	**16.026**
E^{FFHQ}_{var2}	**6.589**	20.081	**4.456**	19.983	19.469	30.181	**4.262**	23.717
E^{Ours}_{var2}	9.829	**16.775**	6.672	**15.047**	**13.022**	**14.836**	6.571	**12.221**

Table 3: FID (↓) for EG3D generators that are trained on different datasets. We calculate the FIDs by sampling 50,000 images using different sampling strategies and different camera distributions. We compare the models that are trained with the same training strategy ($var1/var2$).

EG3D). See the calculated FID values in Tab. 3.

Models trained on our datasets exhibit improvements in FID in most cases, particularly when the final results are rendered from large poses ($c_r \sim LPFF$), or when the generator is conditioned on large poses ($c_g \sim LPFF$). We notice that there is an increased FID when computing $c_g = c_{avg}/c_r, c_r \sim FFHQ$. As explained by the authors of EG3D, the pre-trained E^{FFHQ}_{var1} and E^{FFHQ}_{var2} were achieved using buggy (XY, XZ, ZX) planes. We fix this bug as they suggested using (XY, XZ, ZY), but the XZ-plane representation's dimension is halved, which weakens the expressive capability for frontal faces.

Thanks to our dataset rebalancing strategy, E^{Ours}_{var2} can pay more attention to large pose data and enhance the rendering quality, thus further improving the FID of E^{Ours}_{var1} on large poses. When computing FID of $c_g = c_{avg}, c_r \sim FFHQ$, we notice that E^{Ours}_{var2} has an increased FID compared to E^{Ours}_{var1}, while E^{FFHQ}_{var2} and E^{FFHQ}_{var1} have comparable results. This is due to the addition of new large-pose data, *LPFF*. FID is highly related to the data distribution, and the rebalancd *FFHQ+LPFF-rebal* dataset changes the data distribution when rendering from medium poses.

Facial identity consistency. We leverage ArcFace [8] to measure the models' performance on facial identity maintenance. We render two novel views for 1,024 random faces and use ArcFace to compute the mean identity similarity for all image pairs. We employ three sampling strategies for c_g to evaluate the generator's performance on the camera distribution of different datasets. As for c_r, we find that the extreme rendering camera views will heavily influence the performance of ArcFace, so we only sample c_r from the *FFHQ* dataset, where most of the faces have small to medium poses. As shown in Tab. 1, our models present significant improvements in facial identity consistency across different sample strategies and datasets.

Geometry consistency. We employ $\mathcal{F}$, which outputs 3DMM coefficients to evaluate the geometry consistency. We employ the same camera sampling methods as in facial identity consistency computation. We first render two novel views for 1,024 random faces. Then for each image pair, we compute the mean L2 distance of the face id and expression coefficient. As shown in Tab. 2, our models present improvement in geometry consistency across different sample strategies and datasets.

Image inversion. To evaluate the ability to fit multi-view images, we use FaceScape [48] as the testing data. We use four multi-view images (including one with a small pose) of a single identity as the reference images. We perform latent code optimization to simultaneously project one or four im-

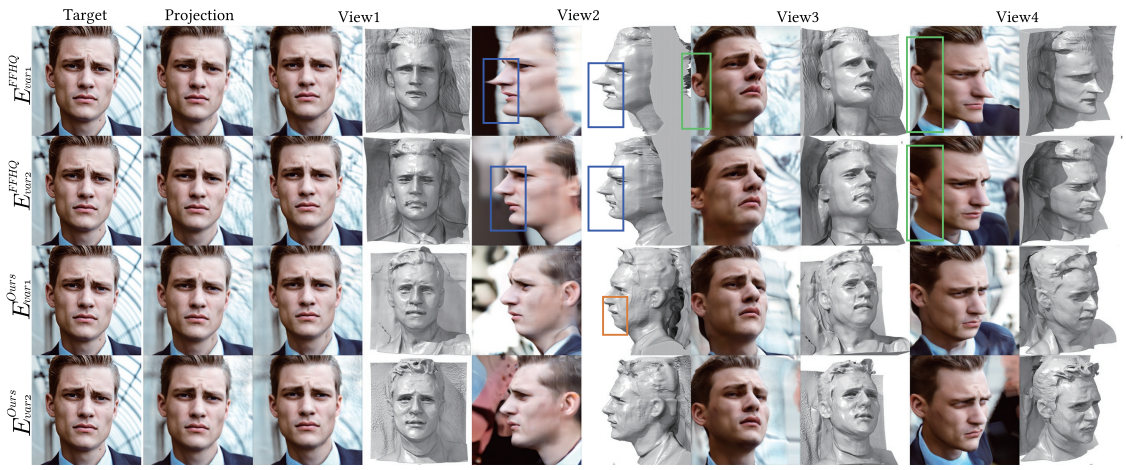


Figure 9: To fit the single-view testing image, we employ HFGI3D [46]. The obtained latent codes are then rendered using four novel views. The inversion is carried out in W space, and the generators are conditioned on c_{avg}.

ages into $W+$ latent space. Then we use the camera parameters that are extracted from another 4 multi-view images to render novel views. Please refer to the supplement for multi-view image inversion results. Because occluded face parts are unavoidable in single-view portraits, we perform single-view image inversion using HFGI3D [46], a novel method that combines pseudo-multi-view estimation with visibility analysis. As shown in Fig. 9, the inversion results indicate that E^{FFHQ}_{var1} and E^{FFHQ}_{var2} suffer from the "wall-mounted" unrealistic geometry. Due to the adhesion between the head and the background, there are missing ears in View 2 and distorted ears and necks in Views 3 and 4 (highlighted by green boxes). A pointed nose exists in View 2 (highlighted by blue boxes). Our E^{Ours}_{var1} and E^{Ours}_{var2} models produce reconstructed face geometry that is free from those artifacts, suggesting that the learned 3D prior from our dataset is more realistic. It also shows that after employing the data rebalance in Sec. 3.4, lips are more natural in E^{Ours}_{var2} compared to E^{Ours}_{var1} (highlighted by an orange box).

"Seam" artifacts. The authors of IDE-3D speculate that the "seam" artifacts in EG3D could be caused by the imbalanced camera pose distribution of datasets, and propose a density regularization loss to deal with the "seam" artifacts along the edge of the faces. Compared to the IDE-3D, our model E^{Ours}_{var1} is trained without requiring any additional regularization loss or any data rebalance strategy, and is free from the "seam" artifacts. Please refer to the supplement for the illustration of "seam" artifacts.

6. Conclusion

In order to address the pose imbalance in the current face generator training datasets, we have presented *LPFF*, a large-pose Flickr face dataset comprised of 19,590 high-quality real large-pose portrait images. Compared to those models trained on *FFHQ*, the 2D face generators trained on our dataset display a latent space that is more representative of large poses and achieve better performance when projecting and manipulating large-pose data. The 3D-aware face generators trained on our dataset can produce more realistic face geometry and render higher-quality results at large poses. The rendering results are also more view-consistent. We hope our dataset can inspire more portrait generating and editing works in the future.

Our work has several limitations. Despite having a more balanced camera pose distribution, our dataset still has a semantic attribute imbalance. For instance, we measured the probability of smiling in *FFHQ+LPFF*. The plot shows that people typically smile when they are facing the camera, so the models trained on our dataset have the smile-posture entanglement. Please refer to the supplement file for the smiling probability plot. This can be overcome by building a camera system and capturing large-scale semantic-balanced images. Our processing pipeline uses the face detector and face reconstruction model to align faces, but it does not perform well under extreme conditions (for example, when only the back of the head is visible and the face is completely occluded). As a result, we cannot obtain full-head results. Future work that can model the full head may be helpful to get a more extensive dataset.

ACKNOWLEDGMENTS Xiaogang Jin was supported by Key R&D Program of Zhejiang (No. 2023C01047) and the National Natural Science Foundation of China (Grant No. 61972344). Hongbo Fu was supported by the Chow Sang Sang Group Research Fund (Project No. 9229119). We extend our sincere gratitude to Sida Peng, whose insightful suggestions significantly improved the quality of this paper. We would also like to thank the Flickr users who shared their portraits for non-commercial use.

References

[1] Rameen Abdal, Hsin-Ying Lee, Peihao Zhu, Menglei Chai, Aliaksandr Siarohin, Peter Wonka, and Sergey Tulyakov. 3davatargan: Bridging domains for personalized editable avatars. *CoRR*, abs/2301.02700, 2023. 2, 3

[2] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3):21:1–21:21, 2021. 1, 2

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR*, 2019. 2

[4] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230, 000 3d facial landmarks). In *IEEE International Conference on Computer Vision, ICCV*, pages 1021–1030, 2017. 3, 4

[5] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 16123–16133, June 2022. 1, 2, 3, 5

[6] 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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5799–5809, 2021. 3

[7] Shu-Yu Chen, Feng-Lin Liu, Yu-Kun Lai, Paul L. Rosin, Chunpeng Li, Hongbo Fu, and Lin Gao. Deepfaceediting: deep face generation and editing with disentangled geometry and appearance control. *ACM Trans. Graph.*, 40(4):90:1–90:15, 2021. 2

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4690–4699, 2019. 7

[9] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. GRAM: generative radiance manifolds for 3d-aware image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10663–10673. IEEE, 2022. 1

[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 *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 285–295, 2019. 3

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1, 2

[12] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *The Tenth International Conference on Learning Representations,ICLR*, 2022. 1, 2, 3

[13] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5767–5777, 2017. 2

[14] Philipp Henzler, Niloy J. Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV*, pages 9983–9992, 2019. 3

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 6626–6637, 2017. 5

[16] Kaiwen Jiang, Shu-Yu Chen, Feng-Lin Liu, Hongbo Fu, and Gao Lin. Nerffaceediting: Disentangled face editing in neural radiance fields. 2022. 1, 2, 3

[17] Wonjoon Jin, Nuri Ryu, Geonung Kim, Seung-Hwan Baek, and Sunghyun Cho. Dr.3d: Adapting 3d gans to artistic drawings. In Soon Ki Jung, Jehee Lee, and Adam W. Bargteil, editors, *SIGGRAPH Asia 2022 Conference Papers, SA 2022, Daegu, Republic of Korea, December 6-9, 2022*, pages 9:1–9:8. ACM, 2022. 2, 3

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR*, 2018. 2

[19] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, volume 33, pages 12104–12114, 2020. 2

[20] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS*, pages 852–863, 2021. 2

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 4401–4410, 2019. 2, 5, 6

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 8107–8116, 2020. 2

[23] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 1867–1874, 2014. 2

[24] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3d GAN inversion with pose opti-

mization. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pages 2966–2975. IEEE, 2023. 2, 3

[25] Martin Köstinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *IEEE International Conference on Computer Vision Workshops, ICCV*, pages 2144–2151, 2011. 3

[26] Thomas Leimkühler and George Drettakis. Freestylegan: free-view editable portrait rendering with the camera manifold. *ACM Trans. Graph.*, 40(6):224:1–224:15, 2021. 4

[27] Peipei Li, Xiang Wu, Yibo Hu, Ran He, and Zhenan Sun. M2FPA: A multi-yaw multi-pitch high-quality database and benchmark for facial pose analysis. *CoRR*, abs/1904.00168, 2019. 3

[28] Connor Z. Lin, David B. Lindell, Eric R. Chan, and Gordon Wetzstein. 3d GAN inversion for controllable portrait image animation. *CoRR*, abs/2203.13441, 2022. 2, 3

[29] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy J. Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020. 3

[30] 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. pages 13503–13513, 2022. 1, 2, 3

[31] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV*, pages 2065–2074, 2021. 2

[32] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR*, 2016. 2

[33] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 896–903, 2013. 3

[34] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 20154–20166, 2020. 3

[35] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. *CoRR*, abs/2206.07695, 2022. 1

[36] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley, 1992. 4

[37] Jie Shen, Stefanos Zafeiriou, Grigoris G. Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 1003–1011, 2015. 3

[38] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 9240–9249, 2020. 2, 6

[39] Zifan Shi, Yujun Shen, Yinghao Xu, Sida Peng, Yiyi Liao, Sheng Guo, Qifeng Chen, and Dit-Yan Yeung. Learning 3d-aware image synthesis with unknown pose distribution. *CoRR*, abs/2301.07702, 2023. 3

[40] Minjung Shin, Yunji Seo, Jeongmin Bae, Young Sun Choi, Hyunsu Kim, Hyeran Byun, and Youngjung Uh. Ballgan: 3d-aware image synthesis with a spherical background. *CoRR*, abs/2301.09091, 2023. 1

[41] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *CoRR*, abs/2206.10535, 2022. 1, 3

[42] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. IDE-3D: interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *ACM Trans. Graph.*, 41(6):270:1–270:10, 2022. 1, 2, 3

[43] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 7662–7672. IEEE, 2022. 1

[44] Feitong Tan, Sean Fanello, Abhimitra Meka, Sergio Orts-Escolano, Danhang Tang, Rohit Pandey, Jonathan Taylor, Ping Tan, and Yinda Zhang. Volux-gan: A generative model for 3d face synthesis with HDRI relighting. In *SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference*, pages 58:1–58:9, 2022. 3

[45] Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Alexander Hypes, Taylor Koska, Steven Krenn, Stephen Lombardi, Xiaomin Luo, Kevyn McPhail, Laura Millerschoen, Michal Perdoch, Mark Pitts, Alexander Richard, Jason Saragih, Junko Saragih, Takaaki Shiratori, Tomas Simon, Matt Stewart, Autumn Trimble, Xinshuo Weng, David Whitewolf, Chenglei Wu, Shoou-I Yu, and Yaser Sheikh. Multiface: A dataset for neural face rendering. In *arXiv*, 2022. 3

[46] Jiaxin Xie, Hao Ouyang, Jingtan Piao, Chenyang Lei, and Qifeng Chen. High-fidelity 3d gan inversion by pseudo-multi-view optimization. *arXiv preprint arXiv:2211.15662*, 2022. 2, 3, 8

[47] Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. GIRAFFE HD: A high-resolution 3d-aware generative model. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 18419–18428, 2022. 3

[48] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: A large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 598–607, 2020. 3, 7

[49] Fei Yin, Yong Zhang, Xuan Wang, Tengfei Wang, Xiaoyu Li, Yuan Gong, Yanbo Fan, Xiaodong Cun, Ying Shan, Cengiz Öztireli, and Yujiu Yang. 3d GAN inversion with facial symmetry prior. *CoRR*, abs/2211.16927, 2022. 2, 3

[50] Stefanos Zafeiriou, George Trigeorgis, Grigorios Chrysos, Jiankang Deng, and Jie Shen. The menpo facial landmark

localisation challenge: A step towards the solution. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 2116–2125, 2017. 3

[51] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 4

[52] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. CIPS-3D: A 3d-aware generator of gans based on conditionally-independent pixel synthesis. *CoRR*, abs/2110.09788, 2021. 1

[53] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS*, pages 118–129, 2018. 3

[54] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 146–155, 2016. 3