PIRenderer: Controllable Portrait Image Generation via Semantic Neural Rendering

Yurui Ren[1]   Ge Li ✉[1]   Yuanqi Chen[1]   Thomas H. Li[2,1]   Shan Liu[3]

[1]School of Electronics and Computer Engineering, Peking University

[2]Advanced Institute of Information Technology, Peking University   [3]Tencent America

{yrren,geli,cyq373}@pku.edu.cn   tli@aiit.org.cn   shanl@tencent.com

Abstract

Generating portrait images by controlling the motions of existing faces is an important task of great consequence to social media industries. For easy use and intuitive control, semantically meaningful and fully disentangled parameters should be used as modifications. However, many existing techniques do not provide such fine-grained controls or use indirect editing methods i.e. mimic motions of other individuals. In this paper, a Portrait Image Neural Renderer (PIRenderer) is proposed to control the face motions with the parameters of three-dimensional morphable face models (3DMMs). The proposed model can generate photo-realistic portrait images with accurate movements according to intuitive modifications. Experiments on both direct and indirect editing tasks demonstrate the superiority of this model. Meanwhile, we further extend this model to tackle the audio-driven facial reenactment task by extracting sequential motions from audio inputs. We show that our model can generate coherent videos with convincing movements from only a single reference image and a driving audio stream. Our source code is available at https://github.com/RenYurui/PIRender.
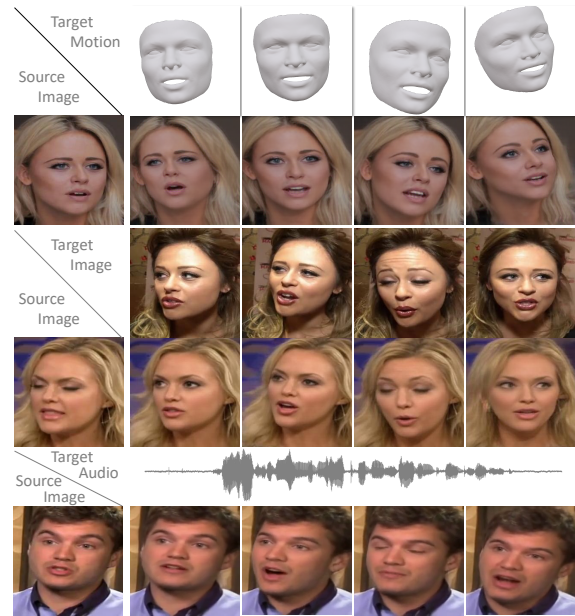
Figure 1. Example results produced by our PIRenderer. This model can generate photo-realistic portrait images according to the user-specified motions (top), motions of another individual (middle), and motions generated from audios (bottom).

1. Introduction

Portrait images are one of the most important photographic depictions that are widely used in daily life. Being able to edit portrait images by intuitively controlling the poses and expressions of given faces (see Fig. 1) is an important task with a large variety of applications in the fields of virtual reality, film production, and next-generation communication. However, enabling such editing is extremely challenging since it requires the algorithm to perceive reliable 3D geometric shapes of given faces. Meanwhile, the acuteness of the human visual system towards portrait images requires the algorithm to generate photo-realistic faces and backgrounds, which makes the task even harder.

Recently, advances in Generative Adversarial Networks (GANs) [14] have made tremendous progress in synthesizing realistic faces [5, 22, 9]. Some methods [3, 42] driven from GANs tackle this task with image translation techniques, where the goal is to train a model such that the conditional distribution of the generated images given input instructions (*e.g.* edges) resembles that of real images. Some follow-up algorithms achieve better generalization by proposing efficient deformation modules [45, 49, 30, 8, 28, 31] or injecting source neural textures to targets [41, 6]. However, the vast majority of existing methods use indirect and subject-specific motion descriptors such as edges [42, 41, 28], semantic segmentation [8], or key-

points [30, 31, 28] to describe the target motions. Although these descriptors with 2D spatial information can benefit the generation of target images, they hinder the model ability to edit portraits in an intuitive manner.

To achieve intuitive control, motion descriptors should be semantically meaningful, which requires facial expressions, head rotations, and translations to be expressed as fully disentangled variables. Parametric face modeling methods [4, 26] provide powerful tools for describing 3D faces with semantic parameters. These methods allow controlling 3D meshes over parameters such as shape, expressions, *etc*. Incorporating priors of these techniques, one can expect to control the generation of photo-realistic portrait images akin to the processing of graph rendering.

In this paper, a neural rendering model PIRenderer is proposed. Given a source portrait image and target 3DMM parameters, our model generates photo-realistic results with accurate motions. As shown in Fig. 2, the proposed model is divided into three parts: the Mapping Network, the Warping Network, and the Editing Network. The mapping network produces latent vectors from motion descriptors. Instructed by the vectors, the warping network estimates deformations between sources and desired targets and generates coarse results by warping sources with the estimated deformations. The editing network generates the final images from the coarse images. The superiority and versatility of our model are demonstrated in the experiments. We show that our model not only enables intuitive image control by editing target images with user-specified motions but also generates realistic results in the indirect portrait editing task where the goal is to mimic the motions of another individual. Additionally, we show the potential of our model as an efficient neural renderer by further extending it to tackle the audio-driven facial reenactment task. Thanks to the high-level fully disentangled parameterizations, we can extract convincing motions from "weak" control audios. Experiments show that our model generates various and vivid motions from an audio stream and transfers the motions into realistic videos of arbitrary target persons. The main contributions are summarized as:

- We proposed a portrait image generation model PIRenderer, which enables intuitive photo-real editing of facial expressions, head rotations, and translations.

- The proposed model can be used to tackle the indirect image editing task which requires imitating other individuals. Thanks to our disentangled modifications and efficient neural renderer, we can extract subject-agnostic motions and generate realistic videos.

- Additional extension on audio-driven facial reenactment demonstrates the potential of the proposed model as an efficient face renderer. Our model generates various and vivid videos from only a single portrait image and a driving audio stream.

2. Related Work

We define portrait editing as the processing of modifying the facial expressions, head rotations, and translations of a given image or video. The existing methods are divided into the following three parts according to the controlling signals.

Portrait Editing via Semantic Parameterization. Editing portrait images using a semantic control space akin to computer animation controls can provide users with intuitive controls. Some model-based methods incorporate the rendering images of 3DMMs and edit portrait images by modifying the expression [38, 39] or pose [23] parameters. These methods achieve impressive results, but they are subject-specific methods, which means that they cannot be applied to arbitrary persons. X2Face [45] achieves better generalization by employing the warping operation to spatially transform source textures. However, it can only edit the pose of source portraits. Recently, generative adversarial networks such as StyleGAN [22] were trained to synthesis realistic face images. Capitalizing on the advantages of these techniques, StyleRig [36] describes a method to control StyleGAN via a 3DMM. However, because it is trained to map latent variables of StyleGAN, it does not support real-world image editing. PIE [35] deals with this shortcoming by proposing an optimization approach to calculate the corresponding embeddings of real images. However, the iterative optimization operation is needed for each input image, which reduces the efficiency of the algorithm.

Portrait Editing via Motion Imitation. Instead of describing target motions with semantic parameters, many models [45, 3, 42, 30, 41, 28, 31, 27, 48] are trained to mimic the motions of another individual. Several methods [3, 46, 42] derived from generative adversarial networks [14] have been proposed to tackle this task by modeling it as the video counterpart of image-to-image translation [19, 52]. These approaches require training on hours of labeled videos of a single person to perform reenactment and therefore have to be retrained for each individual. Some follow-up methods achieve better generalization by proposing efficient spatial transformation modules [30, 31, 28, 43] or injecting source neural textures to targets [49, 6, 48]. These methods can generate realistic images for arbitrary persons. However, they rely on subject-specific (*e.g.* landmarks, edges, parsing maps) or motion-entangled (*e.g.* sparse keypoints) descriptors, which leaves them lacking the ability to intuitively edit source portraits.

Portrait Editing via Audio. Editing portrait images with audios requires generating coherent videos with convincing motions from audio streams and source images. Some methods [10, 40, 32, 51] approach this task by directly modeling the relationships between audio signals and images. However, since audio inputs cannot fully determine the
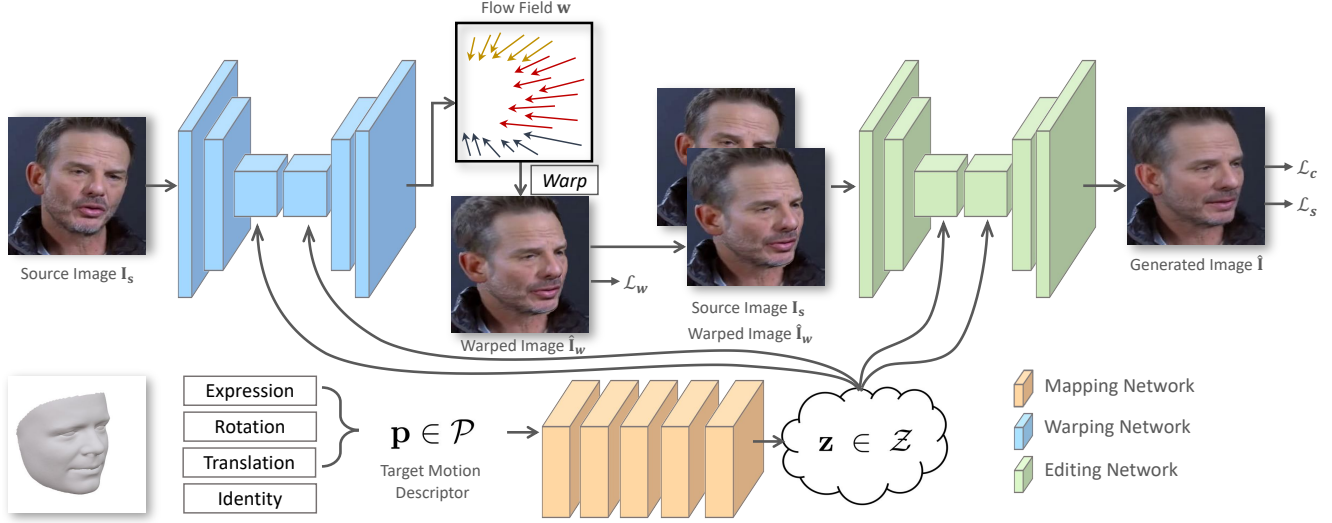
Figure 2. Overview of the PIRenderer. Our model first maps the input motion descriptor **p** to a latent vector **z**. Then a coarse image is generated by the warping network. Finally, the editing network is responsible for generating the final image by editing the coarse result.

head poses of targets, these methods are trained to generate videos with a fixed head pose and therefore do not support full head editing. To avoid interference from the irrelevant factors in real imagery, some models propose to first map audio inputs to intermediate results (*e.g.* landmarks [33, 7], 3DMMs [37, 47, 44]), and then generate the final images. Incorporating the priors of 3DMMs, papers [33, 37, 44, 47] generate targets by synthesizing realistic textures for the rendered images and blending the results with source images. However, these methods require video input and do not work on single images. Meanwhile, they are either subject-specific models [33] or require further fine-tuning before applied to arbitrary individuals [37, 47]. Paper [45] supports full head editing from single images by mapping audio signals to latent variables. However, it does not model the variation of possible motions. Our model can generate various motions for an audio stream and transfer these motions into realistic videos of arbitrary target persons.

3. Our Approach

We propose a semantic neural rendering model PIRenderer for controllable portrait image generation. Instructed by the target motion descriptors **p**, our model can generate photo-realistic portrait image $\hat{\mathbf{I}}$ by modifying the face expressions, head poses, and translations of a given image $\mathbf{I}_s$ while maintaining other source attributes such as identity, illumination, and background. In the following, we first provide the details about the motion descriptors (Sec. 3.1). Then, we introduce our PIRenderer for portrait image semantic control (Sec. 3.2). Finally, we extend our model to tackle a more complex sequential generation task audio-driven facial reenactment (Sec. 3.3).

3.1. Target Motion Descriptor

To achieve intuitive and fine-grained editing, semantically meaningful controls should be provided. In this paper, we employ a subset of 3DMM parameters as the motion descriptor. With a 3DMM, the 3D shape **S** of a face is parameterized as

$$\mathbf{S} = \bar{\mathbf{S}} + \boldsymbol{\alpha}\mathbf{B}_{id} + \boldsymbol{\beta}\mathbf{B}_{exp} \qquad (1)$$

where $\bar{\mathbf{S}}$ is the average face shape, $\mathbf{B}_{id}$ and $\mathbf{B}_{exp}$ are the bases of identity and expression computed via Principal Component Analysis (PCA) based on 200 scans of human faces [26]. Coefficients $\boldsymbol{\alpha} \in \mathbb{R}^{80}$ and $\boldsymbol{\beta} \in \mathbb{R}^{64}$ describe the facial shape and expression respectively. The head rotation and translation are expressed as $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$. With parameter set $\mathbf{p}_i \equiv \{\boldsymbol{\beta}_i, \mathbf{R}_i, \mathbf{t}_i\}$, the desired motions of face i can be clearly expressed.

An off-the-shelf 3D face reconstruction model [12] is employed for extracting the corresponding 3DMM coefficients from real-world portrait images $\mathbf{I}_t$ for training and evaluation. However, coefficient extraction creates the problem of estimation errors. Although the 3D face reconstruction method produces relatively accurate results, errors and noise are inevitable. Mismatches between the extracted motions **p** and the real motions of targets $\mathbf{I}_t$ lead to performance decline and incoherent results (see supplementary materials). To alleviate this problem, the coefficients of a window with continuous frames are used as the motion descriptor of the center frame. Thus, the network can be expected to avoid errors by extracting relationships between adjacent frames. Therefore, the motion descriptor is defined as $\mathbf{p} \equiv \mathbf{p}_{i-k:i+k} \equiv \{\boldsymbol{\beta}_i, \mathbf{R}_i, \mathbf{t}_i, ..., \boldsymbol{\beta}_{i\pm k}, \mathbf{R}_{i\pm k}, \mathbf{t}_{i\pm k}\}$, where k is the radius of the window.

3.2. PIRenderer for Semantic Control

Given a source portrait image $\mathbf{I}_s$ and target motion descriptor $\mathbf{p}$, the PIRenderer generates a portrait image $\hat{\mathbf{I}}$ with accurate target motions while maintaining the other source information such as identity, illumination, and background. Fig. 2 shows an overview of the propositioned architecture. The network combines several components that fulfill specific tasks.

The Mapping Network. We employ a mapping network $f_m : \mathcal{P} \rightarrow \mathcal{Z}$ to produce latent vectors $\mathbf{z} \in \mathcal{Z}$ from the motion descriptors $\mathbf{p} \in \mathcal{P}$.

$$\mathbf{z} = f_m(\mathbf{p}) \qquad (2)$$

The learned latent vectors $\mathbf{z}$ are further transformed by affine transformations to generate $\mathbf{y} = (\mathbf{y}_s, \mathbf{y}_b)$ that control adaptive instance normalization (AdaIN) [18] operations. The AdaIN operations are responsible for injecting the motions described by $\mathbf{z}$ to the warping and the editing network. It is defined as

$$\text{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i} \qquad (3)$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ represent the average and variance operations respectively. Each feature map $\mathbf{x}_i$ is first normalized and then scaled and biased using the corresponding scalar components of $\mathbf{y}$.

The Warping Network. Convolutional Neural Networks lack the capability to spatially transform the inputs in an efficient manner. To better preserve vivid source textures and achieve better generalization, we use a warping network g_w to spatially transform vital information of the source images $\mathbf{I}_s$. The warping network is expected to perceive accurate 3D geometric shapes of source faces and estimates the deformations between input sources and desired targets. It takes the source image $\mathbf{I}_s$ and the latent vector $\mathbf{z}$ as inputs and generates the flow field $\mathbf{w}$ containing the coordinate offsets specifying which positions in the sources could be sampled to generate the targets.

$$\mathbf{w} = g_w(\mathbf{I}_s, \mathbf{z}) \qquad (4)$$

The network is designed with an auto-encoder architecture. AdaIN operation is used after each convolution layer to inject the motions described by $\mathbf{z}$. After papers [34, 31, 28, 21], we do not estimate full resolution flow fields. The output flow is at the $1/4$ resolution of the input image. During training and evaluation, we upsample the predicted flow fields to match the resolution. After obtaining the flow fields $\mathbf{w}$, the coarse results $\hat{\mathbf{I}}_w$ can be calculated by the equation $\hat{\mathbf{I}}_w = \mathbf{w}(\mathbf{I}_s)$ where we warp the source image $\mathbf{I}_s$ with $\mathbf{w}$.

We constrain the warping network for generating accurate flow fields with a warping loss $\mathcal{L}_w$. As the ground-truth

flow fields are not available in this task, we calculate the reconstruction error between the warped image $\hat{\mathbf{I}}_w$ and the target image $\mathbf{I}_t$. The warping loss is based on the perceptual loss of Johnson *et al.* [20] which calculates the ℓ_1 distance between activation maps of the pre-trained VGG-19 network.

$$\mathcal{L}_w = \sum_i \left\| \phi_i(\mathbf{I}_t) - \phi_i(\hat{\mathbf{I}}_w) \right\|_1 \qquad (5)$$

where ϕ_i is the activation map of the i-th layer of the VGG-19 network. Similar to [31], we calculate the perceptual loss on a number of resolutions by applying pyramid down-sampling on $\mathbf{I}_t$ and $\hat{\mathbf{I}}_w$.

The Editing Network. Although the warping network is efficient at spatially transforming the source images, it is limited by the inability to generate contents that do not exist in the sources. Meanwhile, artifacts introduced by the warping operation will lead to performance decline. Therefore, an editing network g_e is designed to modify the warped coarse results $\hat{\mathbf{I}}_w$. The editing network takes $\hat{\mathbf{I}}_w$, $\mathbf{I}_s$, and $\mathbf{z}$ as inputs and generates the final prediction $\hat{\mathbf{I}}$.

$$\hat{\mathbf{I}} = g_e(\hat{\mathbf{I}}_w, \mathbf{I}_s, \mathbf{z}) \qquad (6)$$

The editing network is designed using a similar architecture to that of the warping network. The AdaIN operation is also employed to inject the latent vector $\mathbf{z}$. We use skip-connect layers in the editing network to preserve the source textures. More details of our network architecture can be found in the supplementary materials.

Our editing network is trained with a reconstruction loss $\mathcal{L}_c$ and a style loss $\mathcal{L}_s$. The Reconstruction Loss is used to reduce the errors between the final predictions $\hat{\mathbf{I}}$ and the ground-truth targets $\mathbf{I}_t$. It is designed with similar architecture as that of the warping loss $\mathcal{L}_w$.

$$\mathcal{L}_c = \sum_i \left\| \phi_i(\mathbf{I}_t) - \phi_i(\hat{\mathbf{I}}) \right\|_1 \qquad (7)$$

Similarly, pyramid down-sampling is applied to $\mathbf{I}_t$ and $\hat{\mathbf{I}}$ to calculate the loss on multi-scale images. The style loss $\mathcal{L}_s$ calculates the statistic error between the VGG-19 activation maps as

$$\mathcal{L}_s = \sum_j \left\| G_j^\phi(\mathbf{I}_t) - G_j^\phi(\hat{\mathbf{I}}) \right\|_1 \qquad (8)$$

where G_j^ϕ is the Gram matrix constructed from activation maps ϕ_j. The final loss of the proposed model is a summation of the above losses.

$$\mathcal{L} = \lambda_w \mathcal{L}_w + \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s \qquad (9)$$

In the experiments, we set $\lambda_w = 2.5$, $\lambda_c = 4$, and $\lambda_s = 1000$.
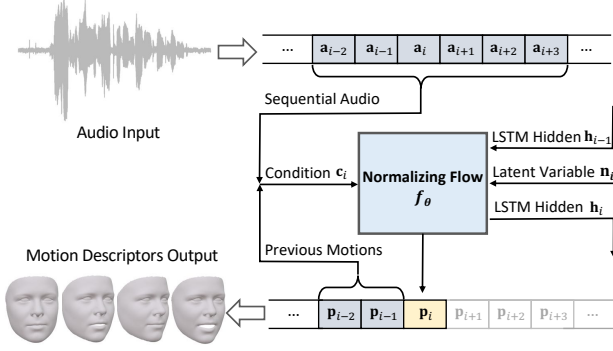
Figure 3. Overview of the audio-driven reenactment model. The previously generated motions $\mathbf{p}_{i-k:i-1}$ and the audios $\mathbf{a}_{i-k:i+\tau}$ are used as the conditional information to generate current motion $\mathbf{p}_i$. Our normalizing flow f_θ is used to map the latent variable $\mathbf{n}_i \in \mathcal{N}$ to the motion descriptor $\mathbf{p}_i \in \mathcal{P}$.

3.3. Extension on Audio-driven Reenactment

In this subsection, we further extend the PIRenderer to tackle the audio-driven facial reenactment task which generates videos with convincing expressions and poses from driving audios and source images. This task requires one to model the relationships between audios and face motions. However, directly mapping audio signals to photo-realistic imagery or other low-level motion descriptors (*e.g.* edges, landmarks) is challenging. Motion-irrelevant factors such as identity, and illumination will interfere with the model. Therefore, employing semantically meaningful parameterizations like 3DMMs as the intermediate results can significantly ease the task. Consequently, we further improve our model by including an additional mapping function f_θ to produce sequential 3DMM coefficients from audios.

A fundamental challenge of designing f_θ is to handle the large variety of possible outputs. Training it to generate deterministic motions will limit the network to produce artifacts as it is constrained to predict the mean motions of all possible results. For this reason, outlining f_θ as a stochastic model that can generate various motions from a single audio stream assists in reducing this issue. *Normalizing flow* [13, 29, 24] is employed to design this model. The core idea of normalizing flow is to train an invertible and differentiable nonlinear mapping function that maps samples from a simple distribution to a more complex distribution. In the conditional setting, the model can be trained to map sequential motion-condition pairs $(\mathbf{p}, \mathbf{c}) \equiv (\mathbf{p}_{1:t}, \mathbf{c}_{1:t})$ to latent variables $\mathbf{n}$ with

$$\mathbf{n} = f_\theta^{-1}(\mathbf{p}, \mathbf{c}) \tag{10}$$

By postulating a simple distribution $p_\mathbf{n}(\mathbf{n})$ (*e.g.* a Gaussian $\mathcal{N}$), the conditional distribution $p_{\mathbf{p}|\mathbf{c}}(\mathbf{p}|\mathbf{c}, \boldsymbol{\theta})$ is obtained by the mapping $\mathbf{p} = f_\theta(\mathbf{n}, \mathbf{c})$ of samples $\mathbf{n} \sim p(\mathbf{n})$.

Specifically, we generate sequential motions in a recurrent manner. Fig. 3 shows the details of the generation of motion $\mathbf{p}_i$ at time i. The previously generated k motions $\mathbf{p}_{i-k:i-1}$ as well as the audios $\mathbf{a}_{i-k:i+\tau}$ are used as the condition information $\mathbf{c}_i$. Meanwhile, in order to extract the temporal correlations, similar to [15, 2], we design the neural network in the normalizing flow layer as a LSTM [17] module. Hidden states are updated at each iteration to generate the current motions. We train this network using a single negative log-likelihood loss. More details about the architecture and the training method can be found in our supplementary materials. In the inference phase, one can generate various sequential motions using $\mathbf{p} = f_\theta(\mathbf{n}, \mathbf{c})$. Then these motions can be transferred into realistic videos using the proposed PIRenderer.

4. Experiment

4.1. Implementation Details

Dataset. We train our model using the VoxCeleb dataset [25] which contains 22496 talking-head videos extracted from YouTube videos. We follow the pre-processing method described in paper [31] to crop faces from the original videos. The cropped videos contain natural movements where faces move freely within a fixed bounding box. The videos are then resized to 256×256 for training and testing. The corresponding audios are extracted from the videos for the audio-driven reenactment task. A total of 17913 training videos and 514 test videos, with lengths varying from 64 to 1024 frames, are obtained.

Evaluation Metrics. *Learned Perceptual Image Patch Similarity [50] (LPIPS)* is used to estimate the reconstruction error. It computes the perceptual distance between the generated images and reference images. Meanwhile, the realism of the synthesized results is measured by *Fréchet Inception Distance [16] (FID)* which calculates the Wasserstein-2 distance between distributions of the fake and real images. The motion accuracy is estimated by the *Average Expression Distance (AED)* and the *Average Pose Distance (APD)* which calculate the average 3DMM expression and pose distance between the generated images and targets respectively. The trained model of Yu *et al*. [12] is used to extract the 3D face coefficients. Besides, *Just Noticeable Difference (JND)* tests are performed to evaluate the subjective quality. Volunteers are asked to choose the more realistic one from the data pair of real and fake samples.

Training Details. We train our model in stages. The mapping network and the warping network are first pre-trained for $200k$ iterations. Then we train the whole model for another $200k$ iteration in an end-to-end manner. We adopt the ADAM optimizer with an initial learning rate as 10^{-4}. The learning rate is decreased to 2×10^{-5} after $300k$ iterations. The batch size is set to 20 for all experiments.
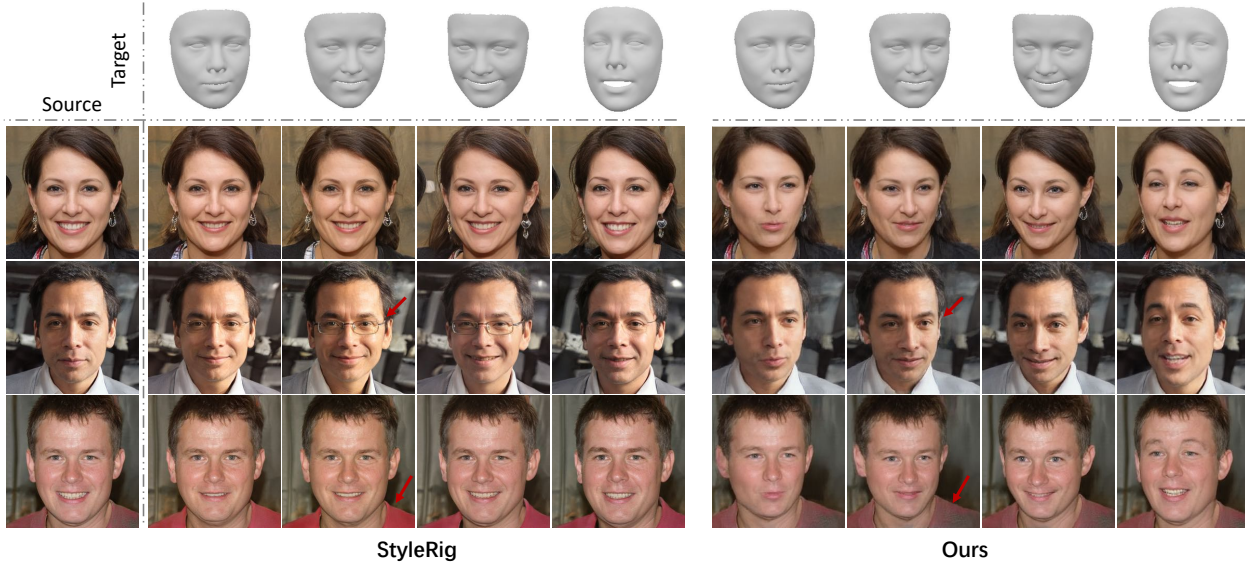
Figure 4. The qualitative comparison results in the intuitive portrait image editing task. The StyleGAN-generated images are used as source images. The left part shows the results of StyleRig [36]. The right part shows the results of our model. We mark the undesired modifications of StyleRig and the corresponding correct results of ours with red arrows.

	FID	AED	APD
StyleRig [36]	**47.37**	0.316	0.0919
Ours	65.97	**0.257**	**0.0252**

Table 1. The evaluation results of the intuitive portrait editing task.

4.2. Intuitive Portrait Image Editing

In this subsection, we evaluate the performance of our model in the intuitive image editing task. Although many methods have been proposed for portrait image semantic editing, few of them enable fully disentangled intuitive motion modification. In this experiment, we compare the proposed model to a state-of-the-art method StyleRig [36]. StyleRig achieves semantic editing by controlling Style-GAN [22] via a 3DMM. Mapping functions are trained to transform the latent variables of StyleGAN according to the provided target motions described by 3DMM coefficients.
Editing of StyleGAN-Generated Images. As StyleRig requires the StyleGAN latent variables to perform the modification, we first conduct a comparison with images generated from StyleGAN. A total of 500 latent variables are randomly sampled to generate the test images with StyleGAN. Target motions are obtained by sampling the motion distributions of the FFHQ dataset [22]. Similar as [11], several VAEs are trained on the motions extracted from the real images of FFHQ. We train these networks to map 3DMM coefficients to Gaussian distributions and then reconstruct the inputs. After training, the VAE encoders are discarded and the decoders are used to describe the motion distributions through Gaussian samples. We randomly sample 100 tar-

get motions with various expressions and rotations [1], which results in $50k$ editing images for each model. The evaluation results are in Tab. 1. It can be seen that by using StyleGAN as the final generator, StyleRig is able to produce more realistic images, which leads to a lower FID score. However, the higher AED and APD scores indicate that it may fail to faithfully reconstruct the target motions. Different from StyleRig, our model can generate images with more accurate movements. Fig. 4 provides quantitative comparisons. It can be seen that StyleRig generates impressive results with realistic details. However, it tends to generate images with a conservative strategy: motions away from the distribution center are weakened or ignored for better image quality. Meanwhile, some motion-irrelevant factors (*e.g.* glasses, cloth) are changed during the modification. Although our model is not trained with FFHQ, it can still achieve impressive results when testing with this dataset. We can generate realistic images with not only correct global poses but also vivid microexpressions such as pouting and eyebrow raising. Additionally, the irrelevant source information is well-maintained.
Editing of Real-World Images. Editing real-world images is a more meaningful task. However, StyleRig cannot be directly applied for this task, since it transforms portrait images at the StyleGAN feature level. Fortunately, techniques such as Image2StyleGAN [1] can embed real-world images into StyleGAN latent space, which can be employed to help with such editing. The editing results

---

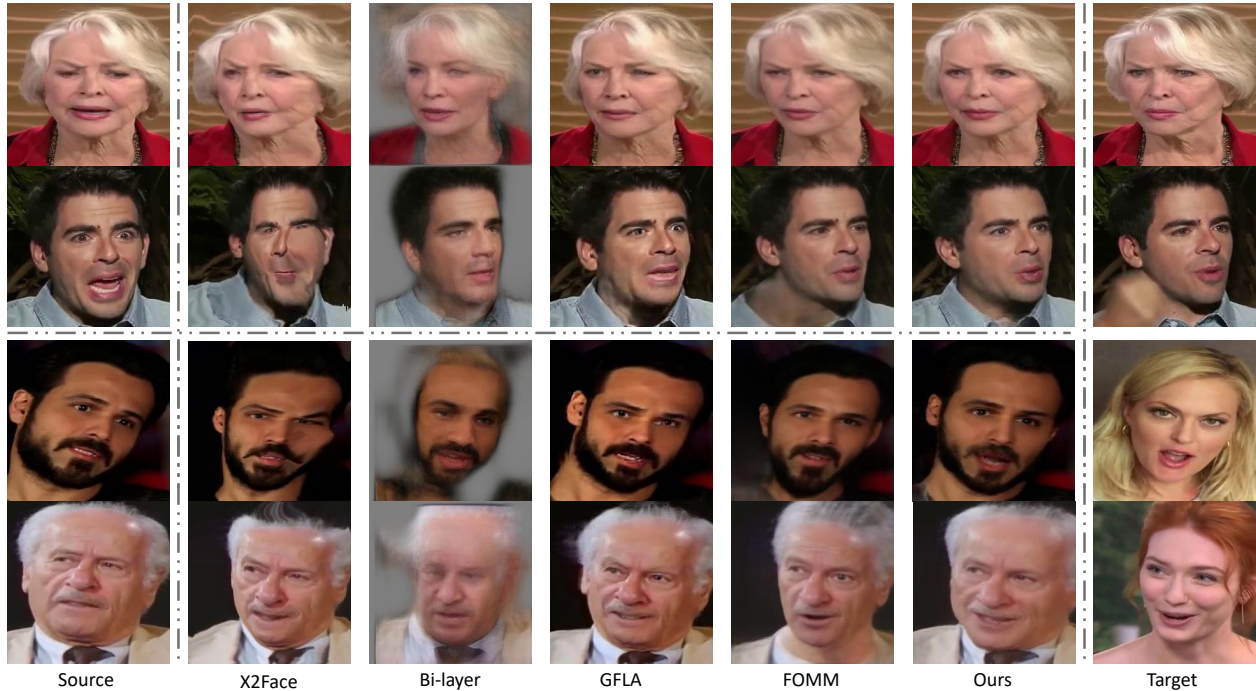[1]We do not edit translations here as StyleRig does not support such editing.

Figure 5. Qualitative comparisons with state-of-the-art methods. The top two lines are the results of the same-identity reconstruction task while the bottom two lines show the results of the cross-identity motion imitation task.
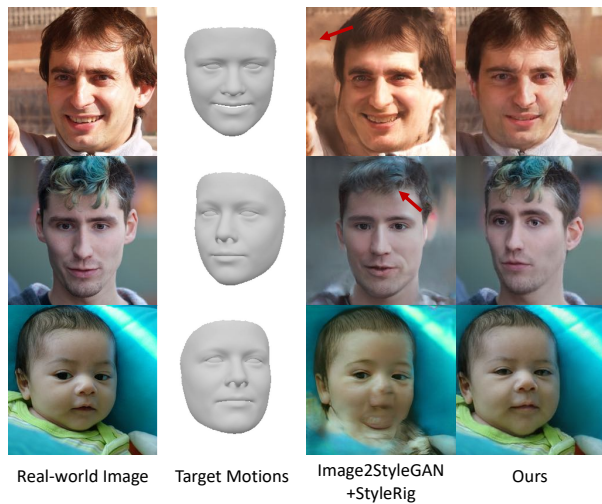

Figure 6. Intuitive editing of the real-world portrait images. Some artifacts are marked by red arrows.

are shown in Fig. 6. Since the optimized embeddings produced by Image2StyleGAN may not obey the original latent distributions, the results of StyleRig are often implausible, which is more evident when complex details are shown in the sources. Our model can be directly applied to edit real-world images. Meanwhile, we can obtain images with fewer artifacts and realistic source textures.

4.3. Talking-head Motion Imitation

In this subsection, we evaluate the performance of our model in the motions imitation task. We compare our model with state-of-the-art methods including: X2Face [45], Bi-layer [48], GFLA [27], and FOMM [31]. The released weights of these methods are used for the evaluation if available. The absolute motions are used for all methods. Since Bi-layer does not predict the background, we subtract the background when doing quantitative analyses.

Two tasks are performed in this experiment: (1) the **Same-identity Reconstruction** task where the source and driving images are of the same person, and (2) the **Cross-identity Motion Imitation** task where non-existing videos are generated by mimicking the motions of another individual. The evaluation results are summarized in Tab. 2. It can be seen that FOMM and GFLA achieve impressive results in the same-identity reconstruction task. However, the performance is reduced when driven by cross-identity videos. One possible explanation is that these methods employ motion descriptors with spatial information (such as sparse keypoints [31] and facial landmarks [27]), which can benefit the reconstruction task. However, the subject-specific motion descriptors may reduce the performance of the cross-identity reenactment task. Our model uses a fully disentangled motion descriptor, which helps with achieving better results in the cross-identity reenactment task. In addi-

	Same-identity Reconstruction					Cross-identity Motion Imitation			
	FID	AED	APD	LPIPS	JND	FID	AED	APD	JND
X2Face [45]	31.75	0.2244	0.1017	0.2400	19.0%	57.74	0.3316	0.1319	5.4%
Bi-layer [48]	71.96	0.2237	0.0449	0.2527	-	83.59	0.2861	0.0494	-
GFLA [27]	12.38	0.1555	0.0268	0.1754	20.3%	32.17	0.2599	0.0363	12.3%
FOMM [31]	9.114	0.1130	**0.0161**	**0.1220**	34.4%	38.92	0.2635	0.0336	19.4%
Ours	**8.260**	**0.1106**	0.0164	0.1285	**36.1%**	**28.18**	**0.2304**	**0.0279**	**25.5%**

Table 2. Comparisons with state-of-the-art methods on talking-head motion imitation. Both same-identity reconstruction and cross-identity motion imitation are conducted to evaluate the performance. The JND scores of Bi-layer are not provided since the absence of the image background will provide clues to volunteers and cause meaningless evaluation results.

tion to the objective metrics, a just noticeable difference test is conducted on Amazon Mechanical Turk (MTurk). Volunteers are asked to choose a more realistic image from image pairs of ground-truth and generated images. We randomly choose 200 images as the test set for each task. Each pair is compared 5 times by different volunteers. The fooling rate is shown in Tab. 2. It can be seen that our model achieves the best scores in both tasks, which means that our model can generate more realistic results.

The qualitative comparisons are provided in Fig. 5. It can be seen that the results of X2Face suffer from the warping artifacts because it directly outputs the warped images. Bi-layer generates images with over-smoothed details, which reduces the realism of the images. GFLA and FOMM can generate photo-realistic results. However, these methods are sensitive to the occlusions in the driving videos. Meanwhile, the source identity cannot be well-preserved when the source face shape is very different from that of the target. The proposed model generates more realistic results along with accurate movements while still preserving the source identity.

4.4. Audio-driven Facial Reenactment

In this subsection, we show that our model can generate coherent videos by extracting convincing motions from audios. Sequential motions with various poses and expressions are first generated by f_θ. Then we transform these motions to arbitrary individuals with our PIRenderer. Qualitative results can be found in Fig. 7. We provide the results of a state-of-the-art method DAVS [51] as a comparison to our model. Evidently, DAVS can extract meaningful lip motions from audios. However, it cannot generate motions that are weakly related to audios such as head poses and eye movements. This inability consequently reduces the realism of the produced videos. Meanwhile, DAVS tackles this task with a deterministic network that cannot model the massive variation of the possible motions. By employing the semantic coefficients as intermediate results, we can generate various motions from a single audio stream and transfer them into realistic videos. It can be seen that our model not only extracts accurate lip motions but it also generates realistic micro expressions such as pout, blinking.

Figure 7. Qualitative results of the audio-driven facial reenactment task. We provide the results of DAVS [51] as a comparison. Our model is able to generate various and vivid motions for an audio stream and transfer the motions into realistic videos.

5. Conclusion

We have proposed PIRenderer, an efficient portrait image neural renderer that is able to control faces via semantically meaningful parameters. Incorporating priors of 3DMMs, our model can conduct intuitive editing of real-world portrait images by modifying the facial expression, head pose, and translation according to user-specified coefficients. Meanwhile, it can also perform motion imitation tasks. Instructed by the subject-agnostic motion descriptors, the model can generate coherent videos with well-maintained source identities. We believe that controlling generative neural networks with flexible graphics permits many exciting applications. Extension on the audio-driven facial reenactment task provides an example and shows the potential of such a combination.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*, pages 4432–4441, 2019.

[2] Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. Style-controllable speech-driven gesture synthesis using normalising flows. *Computer Graphics Forum*, 39(2):487–496, 2020.

[3] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European conference on computer vision (ECCV)*, pages 119–135, 2018.

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

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[6] Egor Burkov, Igor Pasechnik, Artur Grigorev, and Victor Lempitsky. Neural head reenactment with latent pose descriptors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13786–13795, 2020.

[7] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7832–7841, 2019.

[8] Zhuo Chen, Chaoyue Wang, Bo Yuan, and Dacheng Tao. Puppeteergan: Arbitrary portrait animation with semantic-aware appearance transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13518–13527, 2020.

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[10] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. You said that? *arXiv preprint arXiv:1705.02966*, 2017.

[11] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5154–5163, 2020.

[12] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[13] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

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

[15] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[16] 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*, pages 6626–6637, 2017.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[21] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. *arXiv preprint arXiv:2006.04902*, 2020.

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[23] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[24] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pages 10215–10224, 2018.

[25] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[26] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301. Ieee, 2009.

[27] Yurui Ren, Ge Li, Shan Liu, and Thomas H Li. Deep spatial transformation for pose-guided person image generation and animation. *IEEE Transactions on Image Processing*, 2020.

[28] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image

generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7690–7699, 2020.

[29] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[30] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019.

[31] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Advances in Neural Information Processing Systems*, pages 7137–7147, 2019.

[32] Yang Song, Jingwen Zhu, Dawei Li, Xiaolong Wang, and Hairong Qi. Talking face generation by conditional recurrent adversarial network. *arXiv preprint arXiv:1804.04786*, 2018.

[33] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[34] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. *arXiv preprint arXiv:2003.12039*, 2020.

[35] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, Christian Theobalt, et al. Pie: Portrait image embedding for semantic control. *arXiv preprint arXiv:2009.09485*, 2020.

[36] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020.

[37] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *European Conference on Computer Vision*, pages 716–731. Springer, 2020.

[38] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[39] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016.

[40] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. End-to-end speech-driven facial animation with temporal gans. *arXiv preprint arXiv:1805.09313*, 2018.

[41] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. *Advances in Neural Information Processing Systems*, 32:5013–5024, 2019.

[42] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018.

[43] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. *arXiv preprint arXiv:2011.15126*, 2020.

[44] Xin Wen, Miao Wang, Christian Richardt, Ze-Yin Chen, and Shi-Min Hu. Photorealistic audio-driven video portraits. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3457–3466, 2020.

[45] Olivia Wiles, A Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–686, 2018.

[46] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer. In *Proceedings of the European conference on computer vision (ECCV)*, pages 603–619, 2018.

[47] Ran Yi, Zipeng Ye, Juyong Zhang, Hujun Bao, and Yong-Jin Liu. Audio-driven talking face video generation with natural head pose. *arXiv preprint arXiv:2002.10137*, 2020.

[48] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *European Conference on Computer Vision*, pages 524–540. Springer, 2020.

[49] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9459–9468, 2019.

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[51] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9299–9306, 2019.

[52] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.