# OTAvatar: One-shot Talking Face Avatar with Controllable Tri-plane Rendering

Zhiyuan Ma[1,2*] Xiangyu Zhu[3*] Guojun Qi[4] Zhen Lei[1,2,3†] Lei Zhang[1]
[1]The Hong Kong Polytechnic University
[2]Center for Artificial Intelligence and Robotics, HKISI CAS
[3]State Key Laboratory of Multimodal Artificial Intelligence Systems, CASIA
[4]OPPO Research
zm2354.ma@connect.polyu.hk, xiangyu.zhu@nlpr.ia.ac.cn
guojunq@gmail.com, zlei@nlpr.ia.ac.cn, cslzhang@comp.polyu.edu.hk

## Abstract

*Controllability, generalizability and efficiency are the major objectives of constructing face avatars represented by neural implicit field. However, existing methods have not managed to accommodate the three requirements simultaneously. They either focus on static portraits, restricting the representation ability to a specific subject, or suffer from substantial computational cost, limiting their flexibility. In this paper, we propose One-shot Talking face Avatar (OTAvatar), which constructs face avatars by a generalized controllable tri-plane rendering solution so that each personalized avatar can be constructed from only one portrait as the reference. Specifically, OTAvatar first inverts a portrait image to a motion-free identity code. Second, the identity code and a motion code are utilized to modulate an efficient CNN to generate a tri-plane formulated volume, which encodes the subject in the desired motion. Finally, volume rendering is employed to generate an image in any view. The core of our solution is a novel decoupling-by-inverting strategy that disentangles identity and motion in the latent code via optimization-based inversion. Benefiting from the efficient tri-plane representation, we achieve controllable rendering of generalized face avatar at* 35 *FPS on A100. Experiments show promising performance of cross-identity reenactment on subjects out of the training set and better 3D consistency. The code is available at https://github.com/theEricMa/OTAvatar.*

## 1. Introduction

Neural rendering has achieved remarkable progress and promising results in 3D reconstruction. Thanks to the differentiability in neuron computation, the neural rendering methods bypass the expensive cost of high-fidelity digital 3D modeling, thus attracting the attention of many researchers from academia and industry. In this work, we aim to generate a talking face avatar via neural rendering techniques, which is controllable by driving signals like video and audio segments. Such animation is expected to inherit the identity from reference portraits, while the expression and pose can keep in sync with the driving signals.

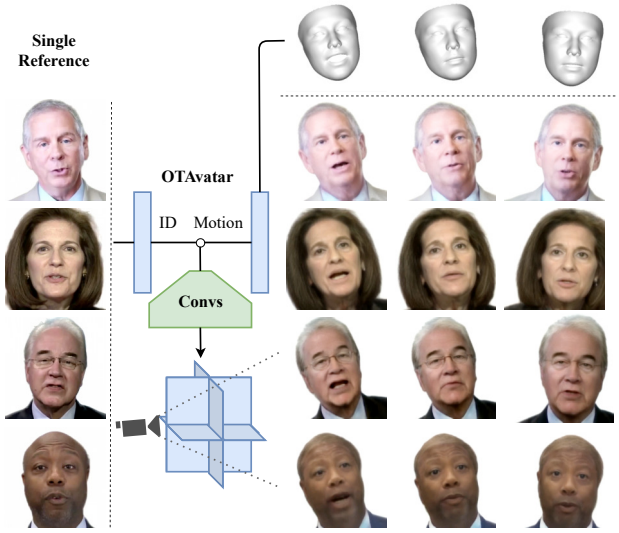


Figure 1. **OTAvatar animation results**. The source subjects in HDTF [43] dataset are animated by OTAvatar using a single portrait as the reference. We use the pose and expression coefficients of 3DMM to represent motion and drive the avatar. Note that these subjects are **not included** in the training data of OTAvatar.

The early works about talking face focus on expression animation consistent with driving signals in constrained frontal face images [5, 7, 27]. It is then extended to in-the-wild scenes with pose variations [18, 35, 46, 47]. Many

*Equal contribution.
†Corresponding author.

previous works are 2D methods, where the image warping adopted to stimulate the motion in 3D space is learned in 2D space [29, 31, 32, 36, 40]. These methods tend to collapse under large pose variation. In contrast, there are 3D methods that can address the pose problem, since the head pose can be treated as a novel view. Both explicit [11, 20] and implicit 3D representations [23] are introduced to face rendering [15, 17, 21, 22, 25, 28, 45]. However, these methods either overfit to a single identity or fail to produce high-quality animation for different identities.

In this work, we propose a one-shot talking face avatar (OTAvatar), which can generate mimic expressions with good 3D consistency and be generalized to different identities with only one portrait reference. Some avatar animation examples are shown in Fig. 1. Given a single reference image, OTAvatar can drive the subject with motion signals to generate corresponding face images. We realize it under the framework of volume rendering. Current methods usually render static subjects [22, 28]. Although there are works [12, 15, 25, 26, 45] proposed to implement dynamic rendering for face avatars, they need one model per subject. Therefore, the generalization is poor. HeadNeRF [17] is a similar work to us. However, its reconstruction results are unnatural in the talking face case.

Our method is built on a 3D generative model pre-trained on a large-scale face database to guarantee identity generalization ability. Besides, we employ a motion controller to decouple the motion and identity in latent space when performing the optimization-based GAN inversion, to make the motion controllable and transferable to different identities. The network architecture is compact to ensure inference efficiency. In the utilization of our OTAvatar, given a single reference image, we fix the motion code predicted by the controller and only optimize the identity code so that a new avatar of the reference image can be constructed. Such a disentanglement enables the rendering of any desired motion by simply alternating the motion-related latent code to be that of the specific motion representation.

The major contributions of this work can be summarized as follows.

- We make the first attempt for one-shot 3D face reconstruction and motion-controllable rendering by taming a pre-trained 3D generative model for motion control.
- We propose to decouple motion-related and motion-free latent code in inversion optimization by prompting the motion fraction of latent code ahead of the optimization using a decoupling-by-inverting strategy.
- Our method can photo-realistically render any identity with the desired expression and pose at 35FPS. The experiment shows promising results of natural motion and 3D consistency on both 2D and 3D datasets.

## 2. Related Works

### 2.1. Talking Face

Regarding the driving signal, the talking face methods can be roughly divided into three categories: audio-driven, image-driven and coefficients-driven. Our approach is most related to coefficients-driven methods, which use either facial landmarks or 3DMM coefficients to represent motion. Facial landmarks involve identity and expression information and is challenging to transfer motion across different subjects. 3DMM coefficients disentangle the identity, expression and pose, and thus they are good driven signals to drive different faces.

Thies et al. [34] used 3D rendering to maintain the shape and illumination attributes when transferring expression and refining mouth details with a mouth retrieval algorithm. Geng et al. [14] used 3DMM to render a given subject with different expressions, followed by a neural network to refine the detail and harmonize with the background. These methods cannot change the face pose. Ren et al. [29] mapped 3DMM expression and pose to a high-dimensional motion representation, then predicted the dense image flow to reenact faces via image warping. Doukas et al. [10] used additional 3DMM mesh fitting results to assist the dense image flow prediction. Yin et al. [40] integrated pre-trained StyleGAN [6] to generate high-resolution talking face prediction by warping low-resolution feature maps. These warping-based methods follow the two-stage workflow by first warping images and then refining facial detail, demonstrating superior performance for same identity reenactment. But in cross-identity reenactment, especially when there is a large pose variation against the source portrait, the animation tends to collapse, since image warping is merely trained to stimulate 3D motion in the 2D space. Our work animates face avatars using 3D rendering and explicitly ensures 3D consistency to handle large pose variations.

### 2.2. Volume Rendering

Volume rendering is a 3D rendering strategy and has demonstrated its success in novel view synthesis. Existing methods either use volume [11, 21] to explicitly represent 3D space or implicitly store the 3D scene in the MLP network [23]. These methods are introduced to animate 3D-consistent portraits. Yu et al. [41] and Raj et al. [28] retrieved the spatial information from sparse support views to render portraits. On the other side, The 3D face generative model can reconstruct a given identity using GAN Inversion. These models vary from pure MLP architecture [4], to low-resolution neural rendering followed by an up-sampling strategy [24, 30], and to using more efficient 3D presentation [3]. All these methods only support rendering static portraits. To animate the avatar with controllable motion, AD-NeRF [15] and NeRFace [13] introduce

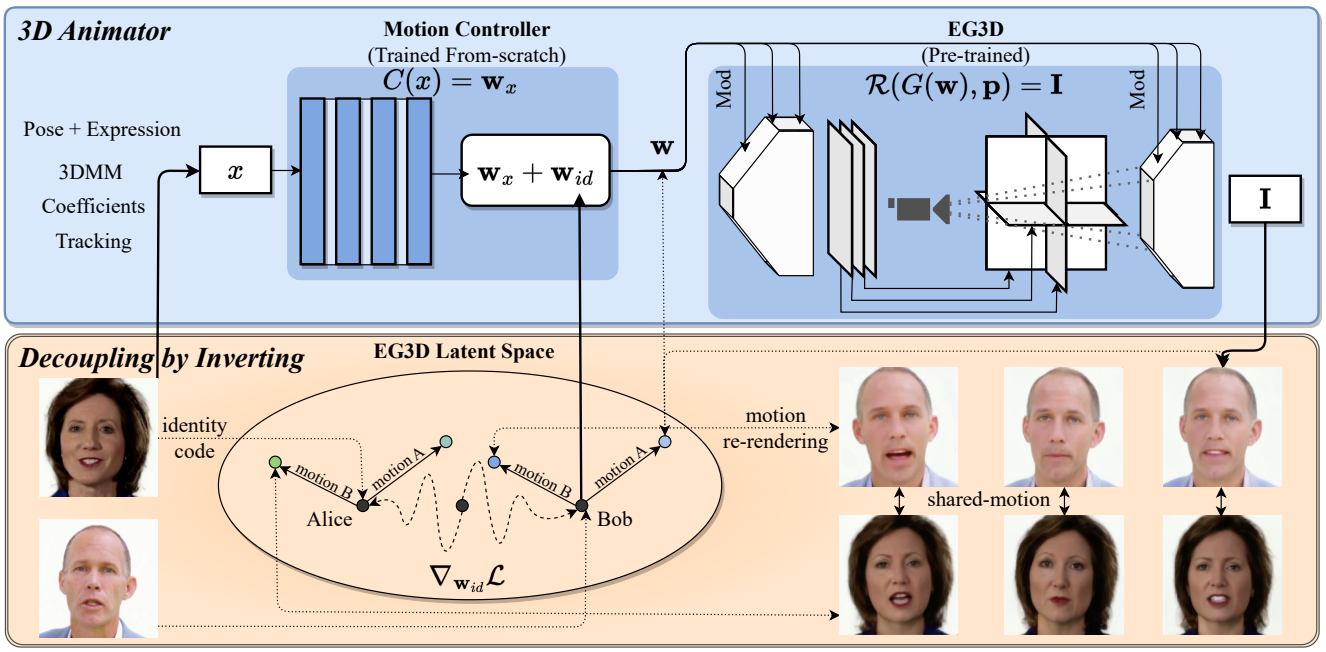


Figure 2. **Overview of OTAvatar.** OTAvatar contains a 3D face animator network and a latent code decoupling strategy, namely decoupling-by-inverting. The 3D face animator is composed of a pre-trained 3D face generator [3] and a motion controller module. The decoupling-by-inverting algorithm is an optimization-based image inversion that can decouple the latent code $\mathbf{w}$ into identity code $\mathbf{w}_{id}$ and motion code $\mathbf{w}_x$. When the model is well-trained, the motion-free identity code can be inferred from a single reference image, and an avatar of an unseen subject can be constructed. The identity code can be integrated with any other motion code predicted by the controller to animate the identity with desired motion.

either the audio feature or 3DMM expression coefficients to the NeRF model to render dynamic heads. Nerfies [25] employs a deformation field conditioned on spatial points to represent the face motion of different frames. HyperNeRF [26] adds an ambient slicing network to tackle the topological drawback of Nerfies. Though these methods have achieved motion control and view consistency for face avatars, they all overfit each NeRF model to one subject, thus lack generalizability. HeadNeRF [17] is trained on both multi-view and frontal face datasets and can be generalized to multiple identities and support motion control using 3DMM coefficients, but its rendering quality is not satisfactory compared with the given portrait, and the motion control is unnatural with random jitters. In contrast, our method can reconstruct photo-realistic face avatars in one shot and animate natural talking motion.

### 2.3. Generative Prior

Many methods rely on pre-trained generative models. Employing a generative model as the prior has the following advantages. First, it provides rich and diverse facial priors, such as texture, color, shape and pose, which can help to restore realistic and faithful facial details. For instance, Yang et al. [39] leveraged a pre-trained GAN model as a prior decoder for blind face restoration. Second, it also enables one-shot image manipulation both inside and outside of the domain. Well-designed schemes for disentanglement are necessary to provide acceptable performance on fine-grained out-of-domain image manipulation. Zhang et al. [44] suggested separating the attributes into global style—like texture and color—and structure—like shape and pose, and performing domain-adaptive generation by transferring the decoupled style attributes. The image manipulation also includes synthesizing human talking videos, Ivan et al. [33] modeled the temporal dynamics of video frames through the latent code disentanglement. Unlike other talking head generation methods [31, 40], it only synthesizes center-aligned faces. The recently presented 3D GANs have been popularized for their ability to generate 3D-consistent photo-realistic human faces with explicit pose control. Katja et.al [30] introduced 3D scene volume rendering without resorting to computationally demanding voxel-based representation, and the model is trained from unposed 2D photos in an adversarial manner. Chan et al. [3] proposed to generate high-quality geometry and multi-view consistent images from 2D photos through a hybrid explicit-implicit network architecture based on StyleGAN2 [19]. Despite the fact that 3D GANs' explicit pose control and 3D-consistent generation are appealing for talking face synthesis, no previous work has studied their application to talking face avatars.

## 3. Method

Talking face avatar aims to synthesize face images with controllable expression and pose. In this paper, we investigate the possibility of endowing volume rendering with the ability to 1) build a faithful identity representation in one shot, 2) enable natural motion control, and 3) achieve real-time inference speed. Our framework takes an identity code and a motion signal as input and generates a tri-plane-based [3] volume through a CNN architecture. By performing volume rendering on the tri-plane representation, where each point is projected on the three feature planes and the sampled features are fused to occupancy and color, a face image at a given camera view can be generated as:

$$\mathbf{I}(\mathbf{p}) = \mathcal{R}(G(\mathbf{w}_{id}, \mathbf{x}; \Theta), \mathbf{p}), \tag{1}$$

where $\mathbf{w}_{id}$ is the identity code, $\mathbf{x}$ is the motion signal, $G(\cdot, \cdot; \Theta)$ is a face volume generation network with $\Theta$ as its network weights to reconstruct an encoded portrait in three orthogonal feature planes, $\mathcal{R}(\mathbf{V}, \mathbf{p})$ is the volume rendering with $\mathbf{V}$ as the volume and $\mathbf{p}$ as the camera view, and $\mathbf{I}(\mathbf{p}) \in \mathbb{R}^{3\times H\times W}$ is the rendered result. In our implementation, the identity code $\mathbf{w}_{id}$ can be extracted by optimizing on a single reference image using our proposed decoupling-by-inverting strategy (Sec. 3.3), achieving one-shot avatar reconstruction. The motion signal $\mathbf{x}$ is 3DMM coefficients for flexible control. We call $G(\cdot, \cdot; \Theta)$ as **3D Animator** because of its ability to animate an avatar with desired motion. Compared with 2D methods, we construct a neural 3D space to promise multi-view consistency. Compared with NeRF, we build a 3D space in the propagation of a CNN, and directly sample features on the tri-plane representation, saving the MLP computation on each sampled point.

### 3.1. 3D Animator Network Structure

**Tri-plane volume representation.** The output of the 3D animator network $G(\cdot, \cdot; \Theta)$ in Eqn. 1 is a tri-plane volume representation, which is composed of three feature planes:

$$\mathbf{V}_{tri} = (\mathbf{F}_{xy}, \mathbf{F}_{xz}, \mathbf{F}_{yz}) = G(\mathbf{w}_{id}, \mathbf{x}; \Theta), \tag{2}$$

where $\mathbf{F}_{xy}$, $\mathbf{F}_{xz}$, and $\mathbf{F}_{yz}$ are three axis-aligned orthogonal feature maps in the 3D space, implicitly forming the volume $\mathbf{V}_{tri}$. When performing volume rendering, for each queried point $(x, y, z)$, we project it onto each of three feature maps and retrieve the corresponding features $(\mathbf{F}_{xy}(x, y), \mathbf{F}_{xz}(x, z), \mathbf{F}_{yz}(y, z))$. The summation of features is sent to a lightweight MLP to decode color and opacity. Compared to fully implicit MLP architectures [4, 15, 45] or volume representation [21], we can efficiently regress the three 2D feature maps through a CNN architecture. In our implementation, $G$ is a deconvolutional network [6] that outputs three $256 \times 256 \times 32$ feature maps.

**Animator structure.** Towards animating the face avatar of a given portrait, an intuitive solution is representing $\mathbf{w}_{id}$ with a pre-defined feature and training $G(\cdot)$ from scratch. However, we find that the model does not work well due to the sub-optimal $\mathbf{w}_{id}$. For example, when employing 3DMM shape coefficients [9] and face recognition features [8], the identity information is not preserved in the rendering results because $\mathbf{w}_{id}$ does not encode face appearances well. These models are trained on cropped facial images which contain no torso and hairstyle. Therefore, $\mathbf{w}_{id}$ losses such information that is necessary when animating portraits. The network even collapses if we make $\mathbf{w}_{id}$ learnable.

In this work, we employ a two-phase strategy to achieve one-shot avatar reconstruction: 1) building a 3D face generator, and 2) making the generator controllable. We build our 3D face generator on a pre-trained EG3D [3] network, which incorporates tri-plane representation for efficient 3D face generation:

$$\mathbf{V}_{tri} = G_{eg}(\mathbf{w}; \Theta_{eg}), \tag{3}$$

where $\mathbf{w}$ is an uninterpretable latent code drawn from the latent space of the generator $G_{eg}$. The latent space is determined by the generator weight $\Theta_{eg}$. To control the latent code $\mathbf{w}$ by the given identity represented by $\mathbf{w}_{id}$ and motion signal $\mathbf{x}$, we propose a motion controller module $C$ parameterized by $\Theta_c$, which maps motion signal $\mathbf{x}$ to the motion code $\mathbf{w}_x$, such that:

$$\mathbf{w} = \mathbf{w}_{id} + \mathbf{w}_x = \mathbf{w}_{id} + C(\mathbf{x}; \Theta_c). \tag{4}$$

Injecting Eqn. 4 to Eqn. 3 , we have the structure of the 3D animator:

$$\begin{aligned} G(\mathbf{w}_{id}, \mathbf{x}; \Theta) &= G_{eg}(\mathbf{w}_{id} + C(\mathbf{x}; \Theta_c); \Theta_{eg}), \\ \Theta &= \Theta_{eg} \cup \Theta_c. \end{aligned} \tag{5}$$

In our implementation, the motion signal $\mathbf{x}$ is constructed by 3DMM pose and expression coefficients [9], which are more flexible compared with image-based driving signals [31, 32, 36], especially in cross-identity driving.

**Controller structure.** We use the concatenation of 3DMM pose and expression coefficients to represent motion $\mathbf{x}$. Practically, we use the adjacent coefficients of pose and expression within a particular radius to represent the motion signal of the current frame, firstly the weighted summation of the coefficients across the temporal dimension is conducted to avoid noises, which is achieved by three 1D convolution layers [29, 40]; secondly, a five-layer MLP is employed to transform the weighted summation into a motion feature; finally, a codebook with learnable orthogonal bases [36] is built, on which each motion feature is projected to get the final motion code $\mathbf{w}_x$. The aforementioned operation is summarized as follows:

$$\mathbf{w}_x = C(\mathbf{x}; \Theta_c) = \mathbf{D} * F_M(F_T(\mathbf{x}; \Theta_T); \Theta_M), \tag{6}$$

where $F_T$ parameterized by $\Theta_T$ is the temporal smoothing network of 1D convolution layers, $F_M$ parameterized by $\Theta_M$ is the five-layer MLP which projects motion to the magnitudes of bases contained in the codebook $\mathbf{D}$. It can be seen that the magnitude projection parameters, temporal smoothing weights, and the codebook are the parameters of the controller $C$ to be trained, that is:

$$\Theta_c = \mathbf{D} \cup \Theta_T \cup \Theta_M. \tag{7}$$

### 3.2. Controller Training

Given a couple of source frame $\mathbf{I}_s$ and driving frame $\mathbf{I}_d$, along with their motion signals $\mathbf{x}_s, \mathbf{x}_d$ and camera views $\mathbf{p}_s, \mathbf{p}_d$, we perform the dual-objective optimization:

$$\begin{aligned} \mathbf{w}_{id}^* &= \arg\min_{\mathbf{w}_{id}} (\mathcal{L}_s + \mathcal{L}_d), \\ \mathcal{L}_s &= \mathcal{L}(I_s, \hat{I}_s) = \mathcal{L}(I_s, \mathcal{R}(G(\mathbf{w}_{id}, \mathbf{x}_s; \Theta), \mathbf{p}_s), \\ \mathcal{L}_d &= \mathcal{L}(I_d, \hat{I}_d) = \mathcal{L}(I_d, \mathcal{R}(G(\mathbf{w}_{id}, \mathbf{x}_d; \Theta), \mathbf{p}_d). \end{aligned} \tag{8}$$

It is implemented to estimate the identity code $\mathbf{w}_{id}$ shared between $\mathbf{I}_s$ and $\mathbf{I}_d$. $G$, $\mathcal{R}$, and $\mathcal{L}$ indicate 3D face generator, volume rendering, and loss function, respectively. The 3D face animator $G$ is parameterized by $\Theta = \Theta_{eg} \cup \Theta_c$. $\Theta_{eg}$ includes the parameters of the pre-trained 3D face generator, and $\Theta_c$ includes the parameters of the controller which is the focus of our training scheme.

**Training scheme**. The purpose of controller training is to decouple the identity code and motion code from the latent code of the generator, making identity and motion replaceable in our generalized avatars. To this end, we propose the *decoupling-by-inverting* training strategy, which is generally an alternation of identity optimization and controller training at different steps. Specifically, we first freeze $\Theta_c$ and perform $N_{id}$ steps of back-propagation on $\mathbf{w}_{id}$. Then we freeze $\mathbf{w}_{id}$ and perform $N_{mo}$ steps of back-propagation on $\Theta_c$ so that the controller learns to add the motion code on top of the identity code. In our implementation, $N_{id} = 90$ and $N_{mo} = 10$. After $N = N_{id} + N_{mo}$ steps, we slightly finetune $\Theta_{eg}$ with a learning rate of $10^{-4}$ to make the rendering fit the driving signal better. A brief pseudo-code of the whole scheme is provided in the supplementary material.

### 3.3. One-shot Avatar Construction

In the inference stage, given a single portrait image $\mathbf{I}_r$, its 3DMM coefficients $\mathbf{x}_r$, and camera view $\mathbf{p}_r$ as the reference of avatar construction, we can estimate the identity code $\mathbf{w}_r$ of the reference portrait via:

$$\mathbf{w}_r = \arg\min_{\mathbf{w}_{id}} \mathcal{L}(\mathbf{I}_r, \mathcal{R}(G(\mathbf{w}_{id}, \mathbf{x}_r), \mathbf{p}_r). \tag{9}$$

Afterwards, we can animate the identity with given driving motion $\mathbf{x}_d$ and camera view $\mathbf{p}_d$, either from the same or different subject, through the following operation:

$$\mathbf{I}(\mathbf{w}_r, \mathbf{x}_d, \mathbf{p}_d) = \mathcal{R}(G(\mathbf{w}_r, \mathbf{x}_d), \mathbf{p}_d), \tag{10}$$

where $\mathbf{I}(\mathbf{w}_r, \mathbf{c}_d, \mathbf{p}_d)$ has the identity estimated as $\mathbf{I}_r$, and is animated with the motion $\mathbf{c}_d$ and camera view $\mathbf{p}_d$. We name Eqn. 9 as the *decoupling-by-inverting* inference strategy since it can estimate decoupled identity code from latent code in the iterative optimization-based GAN Inversion, and can be used to render different motions.

### 3.4. Loss Terms

The loss $\mathcal{L}(\mathbf{I}, \hat{\mathbf{I}})$ in Eqn. 8 and Eqn. 9 measures the error between the animated result $\hat{\mathbf{I}}$ and the ground truth $\mathbf{I}$. First, a pre-trained VGG-19 network is implemented to calculate the distance of multi-scale activation maps $\phi_i(\mathbf{I})$ via:

$$\mathcal{L}_c = \sum_i \left\| \phi_i(\mathbf{I}) - \phi_i(\hat{\mathbf{I}}) \right\|_1, \tag{11}$$

where $\phi_i$ is the $i$-th layer of VGG. Second, the distance between the Gram matrices $G_j^\phi$ constructed from the $j$-th activation maps $\phi_j$ [29] of $\mathbf{I}$ and $\hat{\mathbf{I}}$ is measured:

$$\mathcal{L}_s = \sum_j \left\| G_j^\phi(\mathbf{I}) - G_j^\phi(\hat{\mathbf{I}}) \right\|_1. \tag{12}$$

Third, we utilize the L1 distance performed on the eyes and mouth regions to supervise the expression detail:

$$\mathcal{L}_m = \sum_n \left\| R_n(\mathbf{I}) - R_n(\hat{\mathbf{I}}) \right\|_1, \tag{13}$$

where $R_n$ is either the eyes or mouth region extracted using RoI-align on the bounding boxes calculated using landmarks. Finally, the ID Loss is performed to preserve the identity consistency:

$$\mathcal{L}_{id} = 1 - \cos(E(\mathbf{I}), E(\hat{\mathbf{I}})), \tag{14}$$

where $E$ is the Arcface face recognition model [8].

The animation loss $\mathcal{L}(\mathbf{I}, \hat{\mathbf{I}})$ is a weighted summation of the above loss terms and is implemented in both training and test-time one-shot avatar construction. Additionally, in the training, we maintain the stability of tri-plane representation with monotonic and TV loss [3], and $\mathcal{L}_1$ loss on $\Theta_c$ to avoid the latent code being out of the latent distribution of $G_{eg}$.

## 4. Experiments

### 4.1. Experiment Setting

In this section, we first describe the implementation detail of our proposed method, the dataset used, and the baselines of our work. Then we compare our proposed method with previous 2D and 3D methods on motion controllability and multi-view consistency.

**Implementation details**. We use the off-the-shelf 3D face reconstruction model to extract the expression and pose coefficients of 3DMM [9]. The motion of any timestamp is represented by the window of adjacent 27 frames of expression and pose coefficients. In practice, we find that the extracted head poses contain too much noises to serve as the camera pose. Therefore, we extract the camera pose using the method in [15]. Our model is implemented in Pytorch using four A100 GPUs. The total batch size is 24, with six images per GPU. We use Adam optimizer with a 0.0001 learning rate to train the motion controller $C$ and finetune the generator $G$. Exponential Moving Average (EMA) is employed to update $\Theta_c$ since it can stabilize the training. The model is trained for 2000 iterations, and each iteration contains $N = N_{id} + N_{mo} = 100$ steps to optimize the identity codes and train the motion controller. For each mini-batch of data, a new batch of latent codes are initialized and optimized using the Adam optimizer with a 0.01 learning rate. Both the 64×64 resolution volume rendering results and 512×512 super-resolution results are implemented to calculate the loss. During training, the identity code $\mathbf{w}_{id}$ is optimized in $\mathcal{W}$ space. After being extended to the $\mathcal{W}^+$ space by channel-wisely repeated 14 times, it is combined with the motion code $\mathbf{w}_x$ and then fed into the face generator $G$. It takes less than two days to train the network entirely. During inference, we first optimize the identity code in $\mathcal{W}$ space, then use another $N = 100$ steps to finetune it in $\mathcal{W}^+$ space [1, 2].

**Dataset**. Our model is trained on the HDTF [43] dataset, which contains frontal talking faces from 362 videos and over 300 subjects. Following [40], we use the pre-processing step in [31] and resize images to 512×512 resolution. The test dataset contains 20 videos of subjects out of the training set. To evaluate the novel subject generalizability and controllability in the 3D-consistent fashion, we sample subjects from the Multiface [38] dataset, which is particularly proposed for 3D-consistent face avatars.

**Baseline methods**. We mainly compare our method with previous works that incorporate 3DMM coefficients to reenact face avatars in either 2D/3D fashion and are generalizable to unseen identities. For 2D methods, PIRenderer [29] and StyleHEAT [40] achieve SOTA performance on face reenactment. For 3D methods, HeadNeRF [17] is the most advanced volume rendering method that incorporates controllability and generalizability. Beyond coefficients-based methods, the SOTA image-driven method FOMM [31] is also taken into comparison.

**Evaluation metrics**. We adopt peak-signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [37], and learned perceptual image patch similarity (LPIPS) [42] to evaluate the visual quality. To measure the realism of the synthesized results and the identity preservation, we use frechet inception distance (FID) [16] and cosine similarity of identity embedding (CSIM) [8] based on ArcFace model, respectively. Besides, the average expression distance (AED), average pose distance (APD), and average keypoint distance (AKD) are employed to evaluate the facial expression and pose.

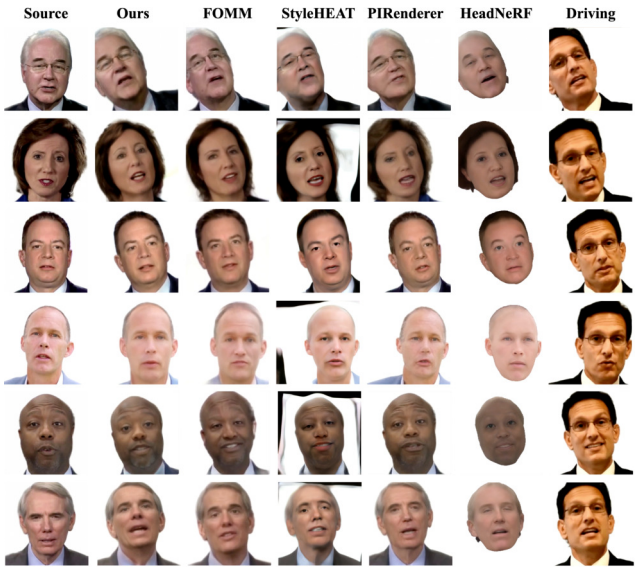


Figure 3. **Qualitative result for cross-identity reenactment**. Examples are sampled from the HDTF dataset [43]. Both source and driving subjects are not included in the training set.

## 4.2. Evaluation Result

We evaluate OTAvatar on animating photo-realistic talking head videos and compare them with state-of-the-art models that support identity-generalized animation methods. In the HDTF [43] dataset, we examine the identity-motion disentanglement by transferring the motion of one subject to drive the other subjects, namely cross-identity reenactment. The employed motion is extracted from a video with large motion variation to evaluate the result in extreme conditions. In the Multiface [38] dataset, we evaluate the consistency of the animation in different views, namely multi-view reenactment. Note that none of the data in this dataset has been used in training our method and baselines. For each talking corpus, we choose the first frame of the frontal-view camera recording as the reference and enforce the network to animate the following frames in frontal view and other views.

**Qualitative comparison**. Fig. 3 shows the results of different methods on cross-identity generation. Compared to FOMM [31], PIRenderer [29] and StyleHEAT [40], which use warping fields for reenacting faces, our method can handle extreme pose variation and maintain identity consistency. Compared to the 3D method of HeadNeRF, our model fully reconstructs the identity-specific detail and syn-


Figure 4. **Qualitative result for multi-view reenactment**. Examples are sampled from the multi-view dataset [38]. All methods use the first frame of the frontal-view portrait to extract the identity feature, and take the expressions of sequential frames and poses of different camera views to generate the talking face. Note that this subject is not included in the training set of any methods.

| | Multi-View Reenactment | | | | | | | | Cross-Identity Reenactment | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | CSIM ↑ | AED ↓ | APD ↓ | AKD ↓ | LIPIPS ↓ | FID ↓ | CSIM ↑ | AED ↓ | APD ↓ | AKD ↓ | FID ↓ |
| FOMM | 20.75 | 0.639 | 0.505 | 2.004 | 0.545 | 5.052 | 0.308 | 101.6 | 0.672 | 3.196 | 0.500 | 4.198 | 113.2 |
| PIRenderer | 20.04 | 0.586 | 0.493 | 2.203 | 0.680 | 6.566 | 0.299 | **100.6** | 0.632 | 3.018 | 0.498 | 4.977 | 103.7 |
| StyleHEAT | 20.03 | 0.632 | 0.387 | 2.179 | 0.472 | 5.522 | **0.284** | 123.8 | 0.614 | 2.860 | 0.471 | 3.592 | 239.1 |
| HeadNeRF | 17.60 | 0.546 | 0.239 | 2.086 | 0.776 | 4.166 | 0.367 | 212.3 | 0.282 | 2.873 | 0.567 | **3.465** | 233.0 |
| Ours | **21.19** | **0.657** | **0.574** | **1.874** | **0.428** | **3.731** | 0.288 | 137.3 | **0.694** | **2.850** | **0.405** | 4.307 | **101.8** |

Table 1. **Quantitative comparisons on multi-View reenactment and cross-identity reenactment.**

thesizes more natural expression. The multi-view consistency results are shown in Fig. 4. First, the 2D warping method suffers facial malformation, which becomes more serious in larger poses. Second, though HeadNeRF renders accurate head poses, the generated results have noticeable deterioration compared to the ground truth. Finally, the face avatar constructed by our method preserves identity details and multi-view consistency.

**Quantitative comparison**. The quantitative comparison among competing methods is shown in Table 1. We can see that the proposed method achieves the best performance in terms of most of the criteria.

**Inference speed**. Table 2 lists the inference speed comparison among state-of-the-art 3D face avatar methods. By utilizing the pre-trained 3D face generator and the compact motion controller (0.8M parameters), our method achieves the highest inference speed, demonstrating its effectiveness.

### 4.3. Ablation Study

**Decoupling-by-inverting**. The controller is trained by the decoupling-by-inverting strategy described in Sec. 3.2, where the training alternates between $N_{id}$ steps of identity code optimization and $N_{mo}$ steps of controller parameters training. To validate the effectiveness of the alternative manner, we discard it and jointly train the controller parameters and the identity code. The qualitative comparison is

| Methods | Frames Per Second ↑ |
|---|---|
| IMAvatar [45] | 0.03 |
| ADNeRF [15] | 0.13 |
| HeadNeRF [17] | 25 |
| Ours | **35** |

Table 2. **Quantitative comparison on the inference speed**. The comparison is conducted on an A100 GPU.


Figure 5. **Qualitative comparison of decoupling-by-inverting training and joint training**. Joint training cannot preserve identity information in one-shot avatar construction.

in Fig. 5. We see that the jointly trained model cannot preserve identity during the animation because the controller manages to overfit the identity information in the training set, and the identity information is not fully encoded in the identity code. The performance degradation caused by joint training is also shown in Table 4. We also analyze the setting of identity optimization step $N_{id}$ and controller training step $N_{mo}$, and list the results in Table 3. We observe that $N_{id} = 90, N_{mo} = 10$ achieves a balance of expression reconstruction (AED) and identity preserving (CSIM).

**Losses**. We also perform an ablation study on the loss functions, including the landmark region losses $\mathcal{L}_m$ and ID losses $\mathcal{L}_{id}$, as shown in Table 4. We can see that the absence of landmark region losses $\mathcal{L}_m$ causes a deterioration of expression reconstruction, measured by AED, while the absence of ID loss causes a performance drop in both expression and identity consistency.

**3D generator finetuning**. When the controller training and identity code optimization are finished, we further slightly finetune the pre-trained EG3D [3] face generator

| $N_{mo}$ | $N_{id}$ | AED ↓ | CSIM ↑ |
|---|---|---|---|
| 1 | 99 | 3.285 | **0.7200** |
| 5 | 95 | 3.178 | 0.716 |
| 10 | 90 | **2.850** | 0.694 |
| 20 | 80 | 3.231 | 0.592 |

Table 3. **Ablation study on the decoupling-by-inverting hyperparameters.** Experiments are conducted under the different settings of $N_{mo}$ and $N_{id}$ with $N = N_{mo} + N_{id} = 100$.

| | AED ↓ | CSIM ↑ |
|---|---|---|
| ours | **2.850** | **0.694** |
| joint | 3.009 | 0.689 |
| w/o $\mathcal{L}_m$ | 3.101 | 0.687 |
| w/o $\mathcal{L}_{id}$ | 3.164 | 0.592 |
| w/o finetune | 3.145 | 0.687 |

Table 4. **Ablation study on the joint training, loss terms, and finetuning**.

with a learning rate of 0.0001. In Table 4, we show the results without finetuning. One can see that slightly finetuning the face generator in the final stage improves identity and expression reconstruction performance.

## 5. Conclusion

We proposed a novel framework of one-shot 3D-consistent talking face avatar, namely OTAvatar, using volume rendering. It jointly addressed the three challenges of face avatars, i.e., generalizability, controllability, and efficiency. For identity-generalized rendering, we implemented a pre-trained 3D face generator to reconstruct faces faithfully, given a single portrait reference. For motion control, we proposed a motion controller module, which predicts the motion code conditioned on 3DMM coefficients. For efficiency, benefiting from the compact architecture of both the 3D generator and controller, our model can animate avatars at a high speed. Besides, we proposed a decoupling-by-inverting approach, which is both a training scheme and a test-time disentangle strategy that decouples the latent code into identity and motion codes via GAN inversion, so that one can animate the avatar with different motions using the motion codes predicted by the controller. Comprehensive experiments were conducted to prove the advantage of our proposed framework.

## Acknowledgement

This work is supported in part by the Chinese National Natural Science Foundation Projects #62176256, #62276254, and the InnoHK program.

## References


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

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8296–8305, 2020.

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

[4] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021.

[5] 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.

[6] Min Jin Chong, Hsin-Ying Lee, and David Forsyth. Stylegan of all trades: Image manipulation with only pretrained stylegan. *arXiv preprint arXiv:2111.01619*, 2021.

[7] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Asian conference on computer vision*, pages 87–103. Springer, 2016.

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

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

[10] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14398–14407, 2021.

[11] Randima Fernando et al. *GPU gems: programming techniques, tips, and tricks for real-time graphics*, volume 590. Addison-Wesley Reading, 2004.

[12] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021.

[13] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, 2021.

[14] Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 3d guided fine-grained face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9821–9830, 2019.

[15] Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5784–5794, 2021.

[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. *Advances in neural information processing systems*, 30, 2017.

[17] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20374–20384, 2022.

[18] Xinya Ji, Hang Zhou, Kaisiyuan Wang, Wayne Wu, Chen Change Loy, Xun Cao, and Feng Xu. Audio-driven emotional video portraits. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14080–14089, 2021.

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[20] Marc Levoy. Display of surfaces from volume data. *IEEE Computer graphics and Applications*, 8(3):29–37, 1988.

[21] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4):65:1–65:14, July 2019.

[22] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. KeypointNeRF: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In *European conference on computer vision*, 2022.

[23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[24] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[25] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[26] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021.

[27] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 484–492, 2020.

[28] Amit Raj, Michael Zollhoefer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pva: Pixel-aligned volumetric avatars. *arXiv preprint arXiv:2101.02697*, 2021.

[29] Yurui Ren, Ge Li, Yuanqi Chen, Thomas H Li, and Shan Liu. Pirenderer: Controllable portrait image generation via semantic neural rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13759–13768, 2021.

[30] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

[31] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 32, 2019.

[32] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13653–13662, 2021.

[33] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022.

[34] 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.

[35] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[36] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. *arXiv preprint arXiv:2203.09043*, 2022.

[37] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[38] 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.

[39] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 672–681, June 2021.

[40] Fei Yin, Yong Zhang, Xiaodong Cun, Mingdeng Cao, Yanbo Fan, Xuan Wang, Qingyan Bai, Baoyuan Wu, Jue Wang, and Yujiu Yang. Styleheat: One-shot high-resolution editable talking face generation via pretrained stylegan. *arXiv preprint arXiv:2203.04036*, 2022.

[41] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[42] 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.

[43] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3661–3670, 2021.

[44] Zicheng Zhang, Yinglu Liu, Congying Han, Tiande Guo, Ting Yao, and Tao Mei. Generalized one-shot domain adaption of generative adversarial networks. *arXiv preprint arXiv:2209.03665*, 2022.

[45] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. I M Avatar: Implicit morphable head avatars from videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[46] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[47] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.