Sparse to Dense Motion Transfer for Face Image Animation

Ruiqi Zhao[1,2], Tianyi Wu[1,2], Guodong Guo[1,2]
[1]Institute of Deep Learning, Baidu Research, Beijing, China
[2]National Engineering Laboratory for Deep Learning Technology and Application, Beijing, China
{zhaoruiqi, wutianyi01, guoguodong01}@baidu.com

Abstract

Face image animation from a single image has achieved remarkable progress. However, it remains challenging when only sparse landmarks are available as the driving signal. Given a source face image and a sequence of sparse face landmarks, our goal is to generate a video of the face imitating the motion of landmarks. We develop an efficient and effective method for motion transfer from sparse landmarks to the face image. We then combine global and local motion estimation in a unified model to faithfully transfer the motion. The model can learn to segment the moving foreground from the background and generate not only global motion, such as rotation and translation of the face, but also subtle local motion such as the gaze change. We further improve face landmark detection on videos. With temporally better aligned landmark sequences for training, our method can generate temporally coherent videos with higher visual quality. Experiments suggest we achieve results comparable to the state-of-the-art image driven method on the same identity testing and better results on cross identity testing.

1. Introduction

We consider the task of face image animation from a single image. Given a source image of a face and a sequence of face landmarks, our goal is to generate a video of the face imitating the motion of landmarks. It has a wide range of applications in video games, filming industry, retails, news broadcasting and teleconferencing among others. It has been a long standing problem of interest in computer graphics. Many works require 3D modelling and a large amount of training data for a specific person [49, 43, 50]. Though capable of generating high quality videos, they have difficulty in generalizing to unseen identities. With the advancement of deep learning techniques, quite a few works [57, 36, 63, 41] are proposed for face image animation when only a few images or even one image of the face is given. Meanwhile, several face video datasets such as VoxCeleb [31, 8, 32] and FaceForensics [38] are collected that facilitate the development of this challenging task.

Recently, Siarohin et al. [41] develops a two-step approach for object-agnostic image animation. It first employs first-order motion model for dense motion estimation and then performs image refinement using an image generator. It is trained in a self-supervised manner and significantly improves the quality of animated face videos. The identity of source face is well maintained and the face motion has lots of details and is temporally coherent. Inspired by the success of [41], we also use the dense motion estimation followed by image generation approach to develop our method. Since [41] assumes local affine motion, affine transformation matrices around a set of keypoints need to be extracted from source/driving image pairs. Different from them, we do not make assumptions on the motion model and only use sparse face landmarks as the driving signal. Our face landmarks are pre-defined and represent rich semantic meaning of the face. We directly conduct motion transfer from sparse face landmarks to the image.

We propose a simple method to transfer motion from sparse face landmarks to the face image. Specifically, we use adaptive instance normalization (AdaIN) layer [25, 21] and our Add_Motion layer as the building blocks to fulfill this goal. AdaIN is invented for image style transfer and has been applied for face image animation [63]. In [63], landmarks are rasterized to an image and AdaIN is used to transfer face identity. Different from them, we concatenate the coordinates of the landmarks and form a low-dimensional vector. We attempt to transfer geometry and motion.

Eyes convey important and engaging message in communication. Therefore it is important to generate gaze change for realistic face image animation. Many existing works do not take it into account in their models [63, 53, 62]. We find it hard to achieve this goal when we apply a single motion transfer network on the whole image. To tackle this issue, we combine a global branch and three local branches in one unified network for motion generation. The global branch looks at the whole image and the local branches each only attends to the left eye, the right eye and the mouth region. The outputs of the four branches are combined to get the final dense motion map.

Being able to generate temporally coherent videos with high visual quality is challenging and it determines the applicability of the derived algorithm. To our best knowledge, existing works that only use landmarks as the driving signal do not consider the influence of detected landmarks quality [63, 53, 62]. We improve face landmark detection on videos by combining a heatmap prediction network [2] with a differentiable regression layer [34] as well as augmenting the regression loss with registration loss (SBR) [12, 11]. We additionally add eye pupils detection to help control gaze.

The contributions of our work are:

- We propose an Add_Motion layer, together with our novel application of AdaIN layer, we are able to transfer motion from sparse landmarks to face images.
- Our global and local motion generation approach allows us to capture global face rotation and translation as well as subtle local motion such as the gaze change.
- We improve face landmark detection on videos and can generate videos with higher visual quality and better temporal coherence.
- Our method achieves comparable results as the state-of-the-art image driven methods on the same identity testing, and better results on cross identity testing.

2. Related Work

2.1. Face Image Animation

Face image animation has long been studied in computer graphics for its wide application in video games and movies [50, 49, 43, 1, 26, 47, 51, 30, 48]. However these algorithms often require RGB-D data of a person, have complicated pipelines and generalize poorly.

Recently several pioneer approaches are developed that do not require personalized training data and can be trained without ground-truth labelling. GANimation [36] builds upon Generative Adversarial Networks (GANs) for the task of facial expression synthesis. It uses a weakly supervised strategy that only requires AU annotations for training. Cycle consistency and attention mechanism are exploited to obtain robust results. X2Face [57] learns to factorize identity and pose/expression, and generate new face image by warping driving pose/expression to the source identity. Instead of warping, [63, 3, 60, 35, 42, 45] generate new face images in a generative adversarial framework. However these methods have to learn frontalized/neutral face as reference and then add pose/expression to generate new images.

Several other works choose to decouple motion and appearance [40, 41, 53, 16, 18, 28, 1, 26, 37]. Monkey-Net [40] is a framework for general object image animation that consists of a landmark detector, a motion prediction network and an image generation network. The landmark detector can learn to detect landmarks in an unsupervised way. It assumes locally rigid motion model surrounding each landmark. First-order [41] assumes locally affine motion model that can model more complex motions. It significantly improves the quality of animated object videos.

2.2. Driving Modalites

The driving modality for face image animation can be face landmarks [10, 13, 53, 63, 20, 62], text [14], audio [46, 43, 5, 66, 9, 65, 7, 52, 4] and images [41, 55]. Suwajanakorn et al. [43] lists a few important practical applications of audio to video generation such as reducing the amount of bandwidth in video transmission, enabling lip-reading from audio for hearing-impaired people and entertainment. Many audio driven face image animation works avoid directly mapping from audio to image, but first convert audio to face landmarks as an intermediate step and then generate images conditioned on synthesized face landmarks [43, 5, 66, 9]. Zhou et al. [66] points out that face landmarks have low degree of freedom and can help bridge the gap between audio and image by representing the facial characteristics. Chen et al. [5] argues that face landmarks can help filter out the noisy signal in audio. Therefore, landmarks driven face image animation is applicable in these audio driven face image animation works.

2.3. Landmarks Driven Face Image Animation

Zakharov et al. [63] learns an identity embedder using a few examples of the source face and combines the rasterized landmarks image and the identity embedded vector in an image generator to synthesize the target image. Their following work [62] largely improves the speed of neural rendering without compromising the visual quality by considering high frequency details. Wang et al. [53] generates a new face image by combining optical flow warped version of the input image and the synthesized intermediate image. Optical flow extracted by FlowNet2 [22] is used as groundtruth for supervision, while we conduct end-to-end learning in an unsupervised manner.

Robust face landmark detection algorithms have been developed over the years. Many existing works [53, 63, 62] use Dlib [27] or FAN4 [2] to detect face landmarks for training their algorithms. However, these landmark detectors do not include eye pupils detection. And when applied on face videos, they return results with lots of temporal jittering. We have added eye pupils detection and used methods to stabilize landmark detection on video. Compared with [53, 63, 62], our method can generate gaze change and our performance is boosted with stabilized face landmarks.

3. Method

Given a source face image and a sequence of driving face landmarks, we would like to generate a face video in which

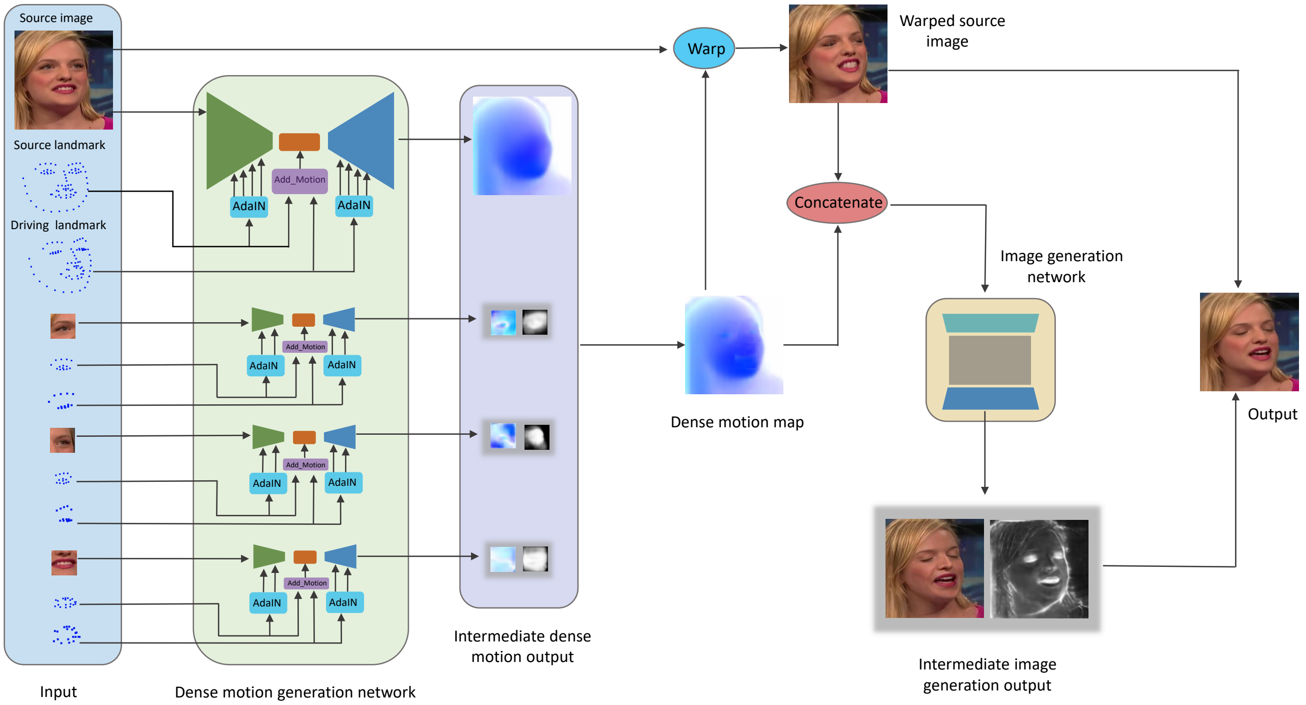

Figure 1. Our method consists of a dense motion generation network and an image generation network. The dense motion generation network contains four branches. The global branch works on the full image. The three local branches each focuses on one region. AdaIN and Add_Motion layers are used for motion transfer in each branch. Outputs of the four branches are combined to get the dense motion map which is used for warping the source image. The warped source image concatenated with the dense motion map is sent to the image generation network to generate a residual image. Warped source image and residual image are weighted and added to get the output image.

the motion of the face resembles the motion of driving face landmarks. Face landmarks are represented using a vector which is the concatenation of coordinates of each landmark. We break down the problem into single face image generation. For each triple of source face image S_I, source face landmarks vector S_{lm} and driving landmarks vector D^k_{lm}, we generate a new face image T^k_I. We then concatenate $T^k_I, k = 1, ..., n$ sequentially to form a face video, where n is the number of frames.

3.1. Method Overview

An overview of our method is shown in Figure 1, where we employ a dense motion generation network and an image generation network in an end to end manner for generating a new face image. The dense motion generation network aims to generate a dense motion map F^k that represents the per pixel mapping. We then warp the source image S_I through F^k and get an initial estimation E^k_I. Using f^b_w to denote the bi-linear warping operation, we have:

$$E^k_I = f^b_w(S_I, F^k) \tag{1}$$

E^k_I and F^k are then concatenated and the result is used as input to the image generation network to synthesize occluded regions and regions that need refinement. With the dense motion map F^k as additional cue, the image generation network better knows where and how E^k_I should be modified and refined. The image generation network outputs a residual image R^k_I and a mask M^k. E^k_I and R^k_I and weighted and added to get the final image:

$$T^k_I = E^k_I \odot (1 - M^k) + R^k_I \odot M^k \tag{2}$$

where $\odot$ is element-wise product. This kind of operation has been employed previously in image animation [17, 40, 41], video-to-video synthesis [54, 53] and image in-painting [61] works.

3.2. Sparse to Dense Motion Transfer

It is challenging to generate dense motion of an image when only sparse motion on a few landmarks is available. We employ AdaIN layer and Add_Motion layer as the building blocks to directly conduct motion transfer from landmarks to image. We further combine global and local motion estimation in a unified network to generate not only the global motion but also local fine motion.

AdaIN layer for motion transfer. Adaptive Instance Normalization (AdaIN) layer has been invented for image style transfer [25, 21, 15]. In [21], AdaIN performs style transfer by aligning feature statistics of content image to match the style image. The feature statistics are channel-wise mean and variance of feature maps. This shows that the feature statistics can encode style information well. In face image animation literature, AdaIN has been used for transferring face identity from source face image to rasterized landmarks image for new face image generation [63].

Different from them, we aim to transfer geometry and motion from sparse landmarks vector to an image. Face landmarks contain rich information about face geometry, including pose, facial expressions and 2D location. By using landmarks vector to control the feature statistics of dense motion generation network through AdaIN, we can make the network carry information of face geometry. Since each feature map is controlled separately with a different mean and variance, different geometry information is transferred to the dense motion network. Our dense motion generation network has an encoder-decoder architecture.

We employ AdaIN to transfer source face landmark geometry to the encoder and driving face landmark geometry to the decoder. Thus, both source and driving face geometry are encoded in the network. Since our training objective is to reconstruct the target image through warping, the dense motion generation network is optimized to learn the change of face geometry and generate a dense motion map.

Add_Motion layer for motion transfer. We propose Add_Motion layer to improve motion transfer from sparse face landmarks to the face image. For Add_Motion layer, we compute the difference between driving face landmarks vector D_{lm}^k and source face landmarks vector S_{lm} and add it to the hidden representation of the dense motion generation network through a fully connected layer. The Add_Motion layer directly injects sparse landmark motion to the dense motion generation network. Through the feed-forward decoder, sparse landmarks motion can be converted to dense pixel motion.

Combine global and local dense motion. AdaIN and Add_Motion layers can transfer satisfying global motion when applied in a dense motion generation network that processes the full image. Without a 3D model, the network can effectively detect the moving foreground and generate face rotation and translation. It can also generate local motion such as eye blinking and mouth opening. However, local motion as subtle as gaze change can not be generated using such one network. Although eyes are only a small part of the face, they can convey important and engaging message. Therefore, it is important to realize gaze change in real practice where we want to make a portrait live.

To tackle this issue, we propose an approach that combines global and local motion estimation. We have a global branch M_G to estimate the global dense motion map, and three small branches M_{L_1}, M_{L_2} and M_{L_3} focus on the left eye eyebrow, the right eye eyebrow and the mouth area, respectively. The input to M_G is S_I of shape 256×256, S_{lm} and D_{lm}^k. The input image to M_{L_1} is a patch of shape 64×64 cropped around the center of left eye eyebrow, and the input source and driving landmarks vector only contains landmarks that belong to the left eye and eyebrow. We do it in the same way for M_{L_2} and M_{L_3}. M_G outputs a dense motion map of size 256×256. M_{L_1}, M_{L_2} and M_{L_3} each outputs a dense motion map and a mask both of size 64×64. The mask represents how to combine global and local motion map in the corresponding region. It helps making the final motion map smoother at the boundary. The local network is better at capturing subtle motion in its local region. Therefore, we are able to get better dense motion maps and generate more realistic images. M_{L_1}, M_{L_2} and M_{L_3} are very small networks that introduce very little computation overhead to the model. The final dense motion map can be denoted as follows:

$$F^k(p) = \begin{cases} f_G^k(p) & p \notin r_{L_i}^k \\ \\ f_G^k(p) \times (1 - m_{L_i}^k(p)) + \\ \hat{f}_{L_i}^k(p) \times m_{L_i}^k(p) & p \in r_{L_i}^k \end{cases} \tag{3}$$

where p denotes a pixel in the image, f_G^k is the dense motion map output of M_G, $r_{L_i}^k, i = 1, 2, 3$ denotes a local region, $m_{L_i}^k$ represents the mask output of M_{L_i}. The dense motion map output of M_{L_i} needs to be shifted and scaled before combing with f_G^k and we use $\hat{f}_{L_i}^k$ to denote the transformed result.

3.3. Improving Face Landmark Detection

Improving face landmark detection on videos is crucial for training a better face image animation model. We would like accurate spatial and temporal alignment of face landmarks sequence. Many existing face landmark detectors, such as Dlib and FAN4 [2], fail to accomplish this goal. We improve face landmark detection on video by combining a heatmap prediction network FAN4 [2] with a differentiable regression layer DSNT [34]. DSNT layer transforms heatmap activation into landmark coordinates. It adds no trainable parameters and is fully differentiable. We additionally augment the regression loss with a registration loss [12, 11] for network training on videos.

Eye pupils convey important information about gaze, therefore we add them in our landmark detection. Our face landmark detector can detect 74 face landmarks.

We initialize our model using pre-trained FAN4 model released by [2]. Since our FAN4 model predicts 74 heatmaps rather than 68 in [2], we randomly initialize other weights that correspond to the 6 new heatmaps. We use the WFLW dataset [58] to train our detector. After the training converges, we further fine tune the detector on videos by augmenting the regression loss with the registration loss.

3.4. Training Losses

Our model is trained on a collection of face videos in an unsupervised manner. We perform face landmark detection on the videos before training. We optionally use a state-of-the-art face parsing algorithm [44] to obtain parsing maps of the videos. During training, we randomly select pairs of source and driving images from each training video. We train the network to reconstruct the driving image D_I^k using the source image S_I, source face landmarks vector S_{lm} and driving face landmarks vector D_{lm}^k.

Our training losses consists of the perceptual loss [41, 23], the least-square GAN loss [29] and a loss that uses face parsing maps as prior. The perceptual loss is given by:

$$L_{per}(D_I^k, T_I^k) = \sum_{j=1}^{J} |N_j(D_I^k) - N_j(T_I^k)| \tag{4}$$

where $N_j(\cdot)$ is the j^{th} channel feature extracted using VGG-19, J is the number of feature channels in VGG-19.

The least-square GAN loss consists of two parts. The first part is for training the discriminator D and is given by:

$$L_{gan}^D(D_I^k, T_I^k) = \|1 - D(D_I^k)\| + \|D(T_I^k)\| \tag{5}$$

The second part is for training the generator G, given by:

$$L_{gan}^G(D_I^k, T_I^k) = \|1 - G(T_I^k)\| \tag{6}$$

We further use face parsing maps as a prior to help guide the training. Given the parsing map for the source image S_P, we warp it using the dense motion map F^k and would like the warped result to match the parsing map of the driving image D_P^k. The warping operation we use is the nearest warping denoted as f_w^n. We compare $f_w^n(S_P)$ with D_P^k and the objective is to minimize the percentage of pixels where the parsing labels are not matched. The loss function is given by:

$$L_{par}(D_I^k, T_I^k) = 1 - |f_{eq}(f_w^n(S_P, F^k), D_P^k)| \tag{7}$$

where f_{eq} is a logic operation function that conducts element-wise comparison of two matrices. It returns 1 if two elements are equal, and 0 otherwise.

The final objective is to minimize the total of all the above losses:

$$L_{total} = L_{per} + L_1 + L_{gan} + L_{par} \tag{8}$$

4. Experiments

We conduct experiments on two datasets, VoxCeleb1 [31, 8, 32] and FaceForensics [38].

VoxCeleb1 is a face video dataset with 22496 videos collected from YouTube. We follow [41] for image preprocessing. During pre-processing, each image is cropped and resized to 256×256. We obtain 12322 videos for training and 512 videos for testing. We use our face landmark detector to extract face landmarks from the videos.

FaceForensics [38] is a much smaller dataset that contains 1004 face videos acquired from YouTube. As in [18] we randomly split the videos into 75% for training and 25% for testing. The videos in FaceForensics are pre-processed in the same fashion as VoxCeleb1.

4.1. Evaluation Metrics

The following metrics [41, 18, 55, 62] are employed in our experiments to quantitatively evaluate our method.

L_1: The average L_1 distance between two images.

PSNR: The mean squared error between two images.

FID score [19]: It measures perceptual realism by extracting feature embeddings using Inception network and computing the average euclidean distance.

SSIM/MS-SSIM [56]: They compare similarity between two images based on luminance, contrast and structure. MS-SSIM is a multi-scale variant of SSIM.

LPIPS [64]: It measures the perceptual similarity between two images by computing cosine distance of each channel and averaging across channels of the conv1-conv5 layers of AlexNet.

AKD: We do face landmark detection using our face landmark detector and compute the average pixel distance. It measures how well the motion is preserved.

4.2. Experimental results

4.2.1 Same identity testing results

We first conduct the same identity testing on VoxCeleb1 and FaceForensics. For VoxCeleb1, we use the same quantitative evaluation setting as [41]. For each testing video, the first frame is used as the source image and all of the frames in the video are used as driving images. We compare our results with the state-of-the-arts, First-order [41], Monkey-Net [40], Few-shot [63] and PuppeteerGAN [6], X2Face [57] in Table 1. We use sparse face landmarks for animating the face image, same for Few-shot and PuppeteerGan, while the other methods use images for driving. Our method performs comparable with First-order and better than all the other methods. Some qualitative results are shown in Figure 2. Our method can generate images with subtle local motion. Using the two landmarks on the pupils, we are able to better control the gaze than First-order while other methods have difficult doing this.

For FaceForensics, we randomly select one image as source image and use all the other images as driving images for each video. We report quantitative comparison results in Table 3. Note that in [18, 57], 16 source images are used while we only use one source image, however, we still achieve better results than them.

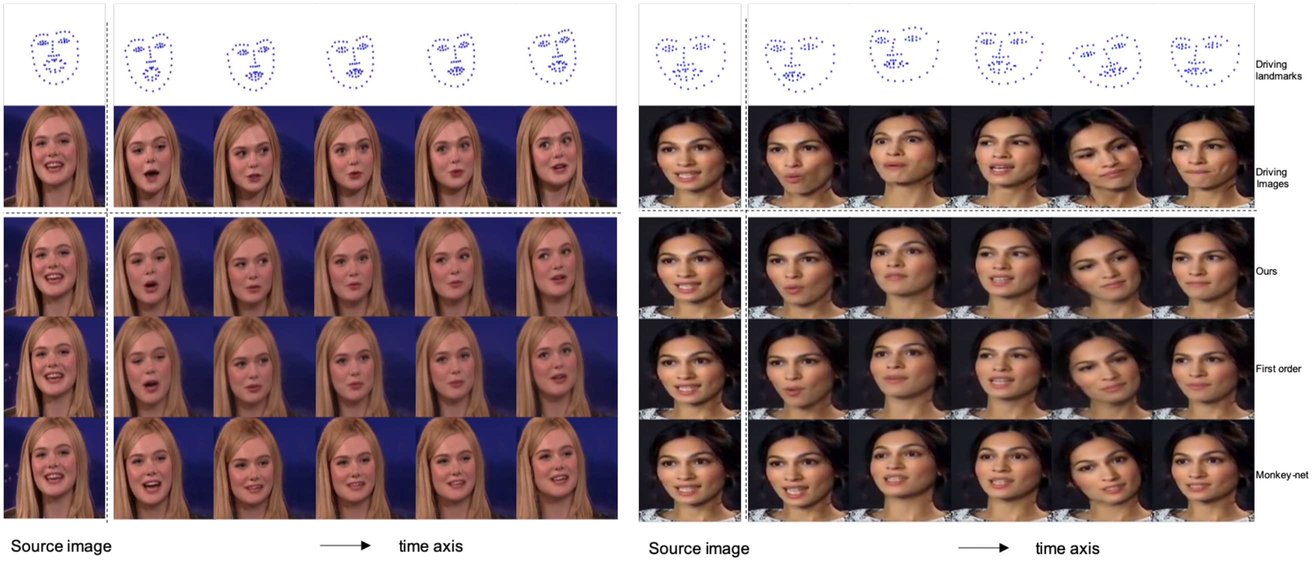

Figure 2. Qualitative results of same identity testing on VoxCeleb1. Our method can generate more accurate gaze change.

Table 1. Comparison with state-of-the-arts on VoxCeleb1 dataset.

Method	L_1 ↓	FID ↓	SSIM ↑	LPIPS ↓	MS-SSIM ↑	AKD ↓	PSNR ↓
Few-shot(2019) [63]	N/A	30.6	0.7200	0.75	N/A	N/A	N/A
PuppeteerGAN (2020) [6]	N/A	33.61	0.7255	N/A	N/A	N/A	N/A
Monkey-Net(2019) [40]	0.0490	12.66	0.7367	0.1249	0.7966	1.76	30.86
First-order(2019) [41]	**0.0415**	10.69	**0.7936**	0.1020	**0.8587**	1.09	**30.97**
Ours	0.0437	**9.06**	0.7770	**0.1004**	0.8438	**0.89**	31.06

Table 2. Comparison of cross identity testing on VoxCeleb1.

Method	FID ↓	User preference ↑
Monkey-net(2019) [40]	70.7	12.2%
First-Order(2019) [41]	57.43	37.8%
Ours	**55.81**	**49.9%**

Table 3. Comparison with state-of-the-arts on FaceForensics.

Method	L_1 ↓	FID ↓
GANimation(2018) [36]	16.19	47.99
X2Face(2018) [57]	11.05	23.98
FLNet(2020) [18]	10.20	20.62
First-Order(2019) [41]	14.06	13.99
Ours	**10.05**	**10.33**

4.2.2 Cross identity testing results

We then conduct cross identity testing. We report quantitative and qualitative cross identity testing results on VoxCeleb1. To do quantitative evaluation on VoxCeleb1, we first select frontal face images in the testing videos. We obtain the 3D landmarks of each face using the 3D face landmark detector [33]. We then get the 3D face pose by aligning the 3D landmarks to a template. We further select images with pose less than 5° for pitch yaw and roll. For each testing video, we get a clip that starts with its selected frontal face image. We select pairs of clips whose starting images are very close in face poses. We use 1° as the threshold. 702 pairs of video clips are obtained for testing.

We compute FID score and compare with First-order [41] and Monkey-Net [40]. The results are shown in Table 2. We achieve a better FID score than them. We also conduct a user study to compare our results with First-order and Monkey-net. In the user study, we randomly select 100 out of the 702 examples in cross identity testing. We ask 18 participants to select the best quality video among the three methods for each example. The results are shown in Table 2. Our results are most preferred among the three methods.

Some qualitative results are shown in Figure 3. When the hairstyles between source face and driving face are very different or when there are occlusions caused by hands, First-order produces more distortion than ours. In Figure 6 we show some failing cases. When there is large pose change or when the face is close to the image boundary, both our method and First-order fail in the eye area.

4.2.3 Cross identity testing of images in the wild

We further perform the challenging images in the wild testing where we download leaders and celebrity images from the Internet and use our model trained on VoxCeleb1 to

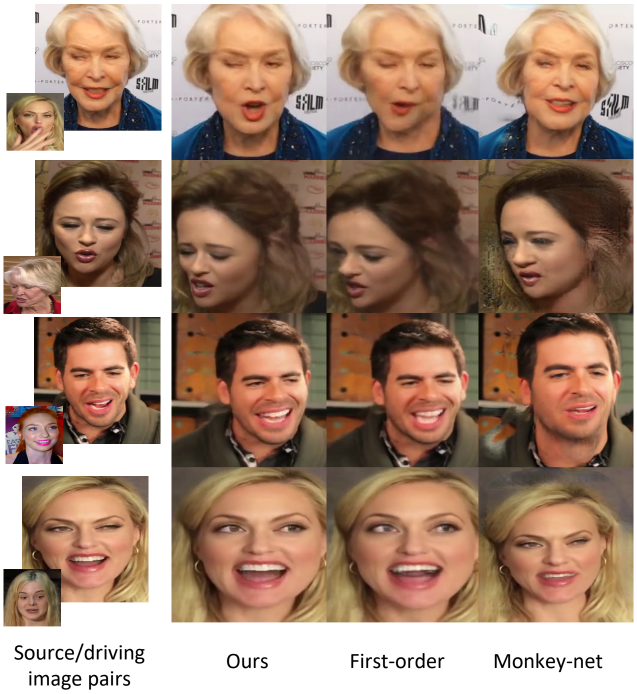

Figure 3. Qualitative results of cross identity testing on VoxCeleb1. Our method is more robust to hairstyle change and occlusion caused by hands. We can also generate images with higher perceptual quality.

generate face images. The testing is also across different identities and we use face image of one person to drive the motion of face image of another person. We compare with First-order and its model is also trained on VoxCeleb1. We show visualization results in Figure 4. From Figure 4, images generated by our method look more realistic with vivid gaze change and have less shape distortion. While images generated by First-order can not follow the direction of the eyes in the driving images.

4.3. Ablation Study

4.3.1 Effectiveness of motion transfer strategies

We conduct ablation study to show the effectiveness of AdaIN layer, Add_Motion layer and the global & local approach for motion transfer. We consider the following three settings. In the first setting, we remove AdaIN layers in the dense motion generation network. In the second setting, we remove Add_Motion layers in the dense motion generation network. And in the third setting, we remove the three local branches in the dense motion generation network. We train and test the three models on VoxCeleb1.

The quantitative comparison results are shown in Table 4. One can see that the full model achieves the best results, showing that all the components contribute to motion transfer in the dense motion generation network. We provide

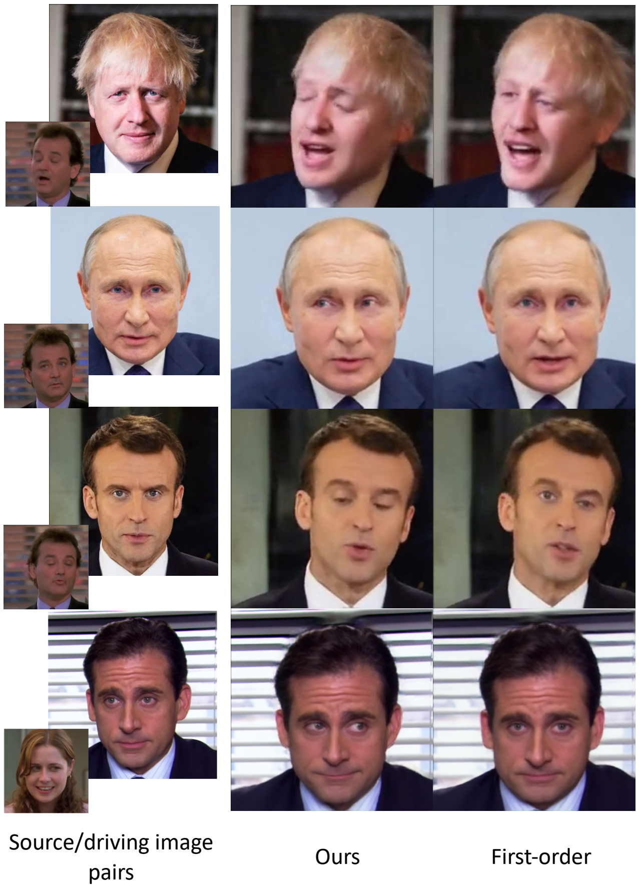

Figure 4. Images in the wild testing. Our method can generate faces with more flexible eyes motion and less shape distortion than First-order.

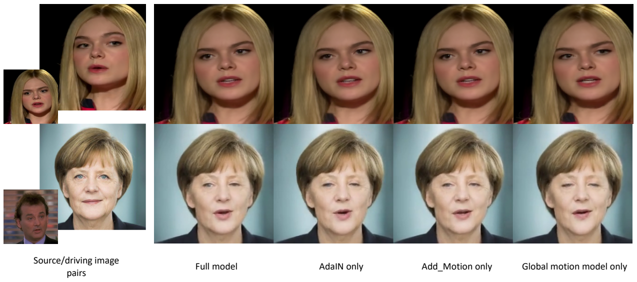

Figure 5. Examples of ablation study on motion transfer. The first example shows that the global motion only model can not generate eye movement. The second example shows that image generated by the full model has higher quality in the eyes and mouth region.

qualitative comparison in Figure 5. Compared with AdaIN only and Add_Motion only model, the full model generates images with higher quality, especially in the eyes and mouth region. Without the local motion model, the global motion only model can not generate eye movement.

Note in this ablation study, we only use warped source image as input to the image generation network without

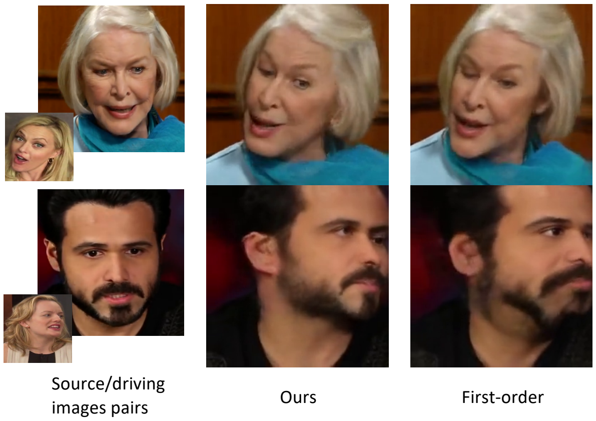

Figure 6. Some fail cases. When the face has large pose change or the face is close to the image boundary, both our method and First-order produce artifacts.

Table 4. Ablation study results on motion transfer.

Setting	$L_1 \downarrow$	FID ↓	SSIM ↑
Full model	**0.0442**	**9.82**	**0.7733**
no Add_Motion layer	0.0449	10.20	0.7728
no AdaIN layer	0.0449	9.98	0.7725
no local motion	0.0454	10.17	0.7729

concatenating it with the dense motion map. Comparing our results in Table 1 and Table 4, we can see that adding dense motion map as additional cue to the image generation network improves the peformance.

4.3.2 Face landmark detector improvement

To show the improvement of our face landmark detector, We compare the four face landmark detectors, Dlib, FAN4, FAN4+DSNT and FAN4+DSNT+SBR. The experiment is conducted on VoxCeleb1 dataset.

We first study their face landmark detection performance on video. To measure temporal smoothness of landmark trajectories, we compute the following metric:

$$SMS = \frac{1}{T}\sum_{t=1}^{T}|(D_{lm}^{t-1} + D_{lm}^{t+1})/2 - D_{lm}^{t}| \qquad (9)$$

where D_{lm}^{t-1}, D_{lm}^{t}, D_{lm}^{t+1} are three consecutive landmark frames, T is the number of frames. We also conduct a user study where we give 10 participants 10 groups of face videos with landmarks marked. We ask the participants to select the video that has the most stable and accurate face landmarks in each group. The videos we use are randomly selected from VoxCeleb1. The results are shown in Table 5. It shows that FAN4+DSNT+SBR is most favored for face landmark detection.

We then train our face image animation model using face landmarks detected by each of the detectors. We use 68 landmarks in this experiment since Dlib and FAN4 only detect 68 landmarks. We only use perceptual loss for training. The performance comparison is shown in Table 5. Using FAN4+DSNT+SBR as the face landmark detector, we are able to boost face image animation performance.

Table 5. Ablation study results on face landmark detector.

Detector	User Study ↑	SMS ↓	$L_1 \downarrow$	FID ↓	SSIM ↑
Dlib	0 %	92	0.0479	12.98	0.7640
FAN4	0 %	115	0.0478	12.71	0.7662
FAN4+DSNT	38 %	61	0.0460	11.88	0.7759
FAN4+DSNT SBR	**62%**	**51**	**0.0457**	**11.70**	**0.7775**

4.4. Limitations

Although we have taken effective ways to improve the quality of generated images, they still look blurry especially when there is large pose change. State-of-the-art super resolution methods [39, 59] may be used to improve the resolution of generated images. Our image generation network has the same architecture as the one in [41]. Techniques from ProGAN [24] and StyleGAN [25] may be borrowed to improve the image generation performance. In addition, the face landmarks we use are very sparse. Some detailed facial motion can not be captured. Denser landmarks could be used to provide more motion guidance. Other modalities such as audio may be used for further improvement.

5. Conclusion

A novel method for face image animation driven by sparse face landmarks has been presented. We adopt the dense motion generation followed by image refinement approach. We propose to use AdaIN layer and our Add_Motion layer for transferring motion from sparse face landmarks to the face image. By combining global and local motion generation, our method can generate not only global motion, such as face rotation and translation, but also fine local motion, such as gaze change. We further improve face landmark detection on video. It greatly improves the visual quality and temporal coherence of generated videos. Experiments have shown that our method achieves comparable results to the state-of-the-art image driven methods on same identity testing and better results on cross identity testing. In the future, we are interested in using the audio modality to further improve the quality of generated face videos.

6. Acknowledgements

Thank Jia Wang for helping downloading and transmitting the VoxCeleb1 dataset. Thank our Reviewer 3 for suggesting adding dense motion map to the image generation network that improves our method.

References

[1] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F Cohen. Bringing portraits to life. *TOG*, 2017.

[2] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *CVPR*, 2017.

[3] Egor Burkov, Igor Pasechnik, Artur Grigorev, and Victor Lempitsky. Neural head reenactment with latent pose descriptors. In *CVPR*, 2020.

[4] Lele Chen, Guofeng Cui, Celong Liu, Zhong Li, Ziyi Kou, Yi Xu, and Chenliang Xu. Talking-head generation with rhythmic head motion. In *ECCV*, 2020.

[5] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *CVPR*, 2019.

[6] Zhuo Chen, Chaoyue Wang, Bo Yuan, and Dacheng Tao. Puppeteergan: Arbitrary portrait animation with semantic-aware appearance transformation. In *CVPR*, 2020.

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

[8] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

[9] Dipanjan Das, Sandika Biswas, Sanjana Sinha, and Brojeshwar Bhowmick. Speech-driven facial animation using cascaded gans for learning of motion and texture. In *ECCV*, 2020.

[10] Xing Di, Vishwanath A Sindagi, and Vishal M Patel. Gp-gan: Gender preserving gan for synthesizing faces from landmarks. In *ICPR*, 2018.

[11] Xuanyi Dong, Yi Yang, Shih-En Wei, Xinshuo Weng, Yaser Sheikh, and Shoou-I Yu. Supervision by registration and triangulation for landmark detection. *PAMI*, 2020.

[12] Xuanyi Dong, Shoou-I Yu, Xinshuo Weng, Shih-En Wei, Yi Yang, and Yaser Sheikh. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In *CVPR*, 2018.

[13] Sefik Emre Eskimez, Ross K Maddox, Chenliang Xu, and Zhiyao Duan. Generating talking face landmarks from speech. In *LVA ICA*, 2018.

[14] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *TOG*, 2019.

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[16] Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. Warp-guided gans for single-photo facial animation. *TOG*, 2018.

[17] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided image generation. 2019.

[18] Kuangxiao Gu, Yuqian Zhou, and Thomas S Huang. Flnet: Landmark driven fetching and learning network for faithful talking facial animation synthesis. In *AAAI*, 2020.

[19] 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 *NeurIPS*, 2017.

[20] Po-Hsiang Huang, Fu-En Yang, and Yu-Chiang Frank Wang. Learning identity-invariant motion representations for cross-id face reenactment. In *CVPR*, 2020.

[21] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[22] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. 2018.

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

[26] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *TOG*, 2018.

[27] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 2009.

[28] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*, 2019.

[29] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

[30] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. pagan: real-time avatars using dynamic textures. *TOG*, 2018.

[31] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*, 2019.

[32] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[33] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[34] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018.

[35] Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitao Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. Realistic dynamic facial textures from a single image using gans. In *ICCV*, 2017.

[36] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*, 2018.

[37] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image generation. In *CVPR*, 2020.

[38] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[39] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.

[40] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *CVPR*, 2019.

[41] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019.

[42] Yang Song, Jingwen Zhu, Dawei Li, Xiaolong Wang, and Hairong Qi. Talking face generation by conditional recurrent adversarial network. *IJCAI*, 2019.

[43] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *TOG*, 2017.

[44] Gusi Te, Yinglu Liu, Wei Hu, Hailin Shi, and Tao Mei. Edge-aware graph representation learning and reasoning for face parsing. In *ECCV*, 2020.

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

[46] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *ECCV*, 2020.

[47] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *TOG*, 2019.

[48] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Real-time expression transfer for facial reenactment. *TOG*, 2015.

[49] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016.

[50] Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Niessner. Headon: Real-time reenactment of human portrait videos. *ACM Transactions on Graphics (TOG)*, 2018.

[51] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popovic. Face transfer with multilinear models. In *TOG*. 2005.

[52] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *IJCV*, 2019.

[53] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. *NeurIPS*, 2019.

[54] 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.

[55] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *CVPR*, 2021.

[56] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

[57] Olivia Wiles, A Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *ECCV*, 2018.

[58] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, 2018.

[59] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P Allebach, and Chenliang Xu. Zooming slowmo: An efficient one-stage framework for space-time video super-resolution. 2021.

[60] Zhiliang Xu, Xiyu Yu, Zhibin Hong, Zhen Zhu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Facecontroller: Controllable attribute editing for face in the wild. *AAAI*, 2021.

[61] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[62] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *ECCV*, 2020.

[63] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *ICCV*, pages 9459–9468, 2019.

[64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[65] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *AAAI*, 2019.

[66] Yang Zhou, DIngzeyu Li, Xintong Han, Evangelos Kalogerakis, Eli Shechtman, and Jose Echevarria. Makeittalk: Speaker-aware talking head animation. *TOG*, 2020.