HeadGAN: Video-and-Audio-Driven Talking Head Synthesis

Michail Christos Doukas[1,2], Stefanos Zafeiriou[1,2], Viktoriia Sharmanska[1]
[1]Imperial College London, London, UK
[2]Huawei Technologies UK, London, UK

Figure 1: *HeadGAN* transfers the facial expressions and head pose from a driving frame to a single reference image. Here, both the source and target identities have not been seen by our model during training.

Abstract

Recent attempts to solve the problem of talking head synthesis using a single reference image have shown promising results. However, most of them fail to meet the identity preservation problem, or perform poorly in terms of photo-realism, especially in extreme head poses. We propose HeadGAN, a novel reenactment approach that conditions synthesis on 3D face representations, which can be extracted from any driving video and adapted to the facial geometry of any source. We improve the plausibility of mouth movements, by utilising audio features as a complementary input to the Generator. Quantitative and qualitative experiments demonstrate the merits of our approach.

1. Introduction

Visual data synthesis [38, 37], including talking head animation [39, 42, 41, 13, 27, 28, 17] are particularly exciting and thriving research areas, with countless applications in games, social media, VR, teleconference and virtual assistance. Over the past years, solutions were mainly given by the graphics community. For instance, *Face2Face* [33] performs face reenactment, recovering facial expressions from a driving video and overwriting them to the source frames. Recent learning-based approaches [17, 19] have sought to solve the problem of full head reenactment, which aims to transfer not only the expression, but also the head pose of the driving person to the source identity. The shortcoming of such methods is their dependence on long video

footage of the source subject, as they train person-specific models. At the same time, various methods have been proposed for reenacting a human head, under a few-shot setting [39, 37, 41, 13, 28], where only a limited number of reference images are available, even a single one. Most state-of-the-art approaches use facial landmarks to guide synthesis [42, 41, 37], which usually leads to identity preservation problems, especially when the facial geometry of the source is dissimilar to the person's appearing in the driving sequence. Furthermore, extreme head poses constitute a notable challenge for most systems [39, 27, 28], as photo-realism seems to diminish substantially for head positions distant from the one appearing in the reference image(s). Last but not least, the mouth movements generated by those methods, are usually quite unnatural and non-plausible, failing to reflect the speech of the driving video.

We propose *HeadGAN*, a novel one-shot head reenactment method. During inference, our system is able to reenact any source image using any driving video, both unseen during training. *HeadGAN* overcomes many limitations of the previous works, owing to the following contributions:

- Our Generator is driven by *a 3D face representation*, extracted from any driving person and then adapted to the identity characteristics of any source subject.
- We propose a network that predicts a *dense flow field*, to map the reference image pose to the desired pose. Different from optical flow, operating on consecutive frames, this unconventional flow enables the Generator to learn any (even extreme) head pose variations.
- Apart from 3D facial information, we condition the generative process on speech features from the audio signal, enabling our method to perform more accurate mouth synthesis.

We perform extensive comparisons with state-of-the-art methods [39, 37, 41, 28] and report comparable or superior performance, in terms of standard GAN metrics [15, 35], even when compared to models [41] trained on the larger VoxCeleb2 [8] dataset. Qualitative experiments suggest an advantage of our model when synthesising "hard" poses. Lastly, we conduct an ablation study, in order to demonstrate the contribution of each component of our system.

2. Related Work

Model-free methods for face synthesis. *X2Face* [39] was among the earliest learning-based methods for animating human heads, which does not rely on any prior knowledge of faces. The authors proposed a warping-based technique for synthesising images with the identity, background and hair coming from the source frame(s), and the poses and expressions taken from the target frames. Despite the photo-realistic generated samples, in many cases the warping operation causes unnatural head deformations. *MonkeyNet* [27] is a deep learning framework that proposes to infer motion via detecting the key-points from the driving video. The appearance extracted from the target image along with motion information is used to generate the output frames. In their recent follow-up work, Siarohin *et al*. [28] significantly improved the results on single image animation. Both of these methods [27, 28] are designed to transfer motion appearing in the driving video, thus the reenactment result depends on the initial head pose appearing in the target image. On the contrary, we propose a head reenactment system that generates the exact same head pose with the one appearing in the source video.

Landmark-based face modeling and generation. There is a plethora of works focusing on the problem of head animation that rely on sparse facial landmarks to drive the synthesis. *Bringing Portraits to Life* [1] was an early attempt to animate still images, where 2D warps are applied on the target image in order to imitate the facial transformations in the driving video. It shows promising results when the source head pose is close to the one appearing in the target image and only a small deformation is required. Much follow-up research on head animation assumed a few-show setting, where a small number of target images are available. Zakharov *et al*. [42] extracts identity related embeddings from the target images and injects them to the generator through adaptive instance normalisation layers (AdaIn) [16]. Their image-based method performs best after fine-tuning on the new target identity. In their most recent work, Zakharov *et al*. [41] propose an one-shot model that capitalises on SPADE [24], while operating in real-time speeds during inference. Using the SPADE layers [24] in order to adapt the generative process to the appearance of the target identity has been proposed earlier in the *Few-shot vid2vid* model [37]. This is a video-based method extending *vid2vid* [38]. By design, all the aforementioned methods are not able to address the identity preservation problem, since facial landmarks allow identity related information of the source person to be transferred into the generated target. *MarioNETte* [13] tries to solve this problem by proposing a method of landmark transformation that adapts the landmarks of the driver to that of the target, in an unsupervised manner. In spite of the attempts to deal with the target identity preservation problem, in principle facial landmarks do not provide a reliable means of transferring information from a source video to a target face, due to their limited representational capacity (*e.g.* 68 points).

Head animation assisted by 3DMMs. On the other hand, *3D Morphable Models (3DMMs)* [2, 4, 3, 5] have been proven to be very effective at modeling human faces and have been widely used to drive face synthesis [33, 17, 29, 32]. Fitting 3DMMs on facial images, enables recovering accurate pose and expressions from the source frames, as well as identity related parameters required from the target

image(s). The rendered 3D face is used to condition a neural network, which completes the texture and fills in the areas of missing information (hair, body, background, etc.). *Deep video portraits (DVP)* [17] and *Head2Head* [19] are examples of such head reenactment systems driven by 3D information. Both methods train *person specific* models, using a long video footage of the target speaker. On the contrary, our proposed approach is *person generic*, *i.e.* it can perform video synthesis for any unseen target speaker. It requires only a single image of the target speaker (identity) and no fine-tuning.

Audio-driven head synthesis. Apart from the video-driven techniques discussed above, there exists an extensive body in literature on audio-driven talking face synthesis [31, 7, 29, 6, 36, 32]. Most of these works fall into two categories. The first one includes methods that receive an audio signal coupled with a still image of the target and generate a face animation corresponding to the audio data. Chung *et al*. [7] proposed an encoder-decoder CNN architecture that uses a joint embedding of the face and audio to create frames. Chen *et al*. [6] designed a GAN-based framework that hallucinated facial landmarks as an intermediate representation and was trained with a dynamically adjustable pixel-wise loss. Song *et al*. [29] proposed a recurrent adversarial network with a lipreading and spatio-temporal discriminator. Vougioukas *et al*. [36] animate still images with audio with an end-to-end GAN model, by taking particular care of the eye region. In all of those approaches, the head pose is mostly fixed when talking. The second category of audio-driven methods consists of facial reenactment systems. Suwajanakorn *et al*. [31] learned to synthesise Obama by generating the mouth area with LSTMs and place it on top of the existing video footage. *Neural voice puppetry* [32] is a recent reenactment system that learns a mapping from the audio features to the expression blend-shapes. The latter ones are then used to render the mouth area with a rendering neural network. Different from the aforementioned works, our system uses audio data signal to enhance realism and speech quality within the mouth area, while head pose and expressions are guided by the source video footage.

3. Methodology

3.1. 3D Face Representation

In order to accurately transfer the expressions of the driving speaker, while preserving the facial geometry of the source identity, we take advantage of prior knowledge on human faces contained within 3DMMs [2, 4, 3, 5]. Given a driving video of T frames, $\mathbf{y}_{1:T} = \{\mathbf{y}_t \mid t = 1, \dots, T\}$, the 3DMM fitting stage produces a sequence of *camera parameters* $\mathbf{c}_{1:T}$ and *shape parameters* $\mathbf{p}_{1:T}$, with $\mathbf{p}_t = [\mathbf{p}_t^{exp\top}; \mathbf{p}_t^{id\top}]^\top$. That is, for each frame t, we obtain two types of shape parameters: a) identity related parameters

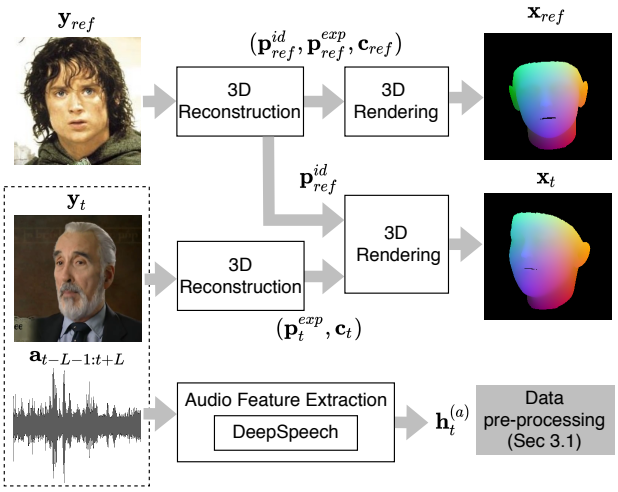


Figure 2: Our pre-processing stage. We reconstruct and render the 3D face of the reference image $\mathbf{y}_{ref}$, as well as the driving frame $\mathbf{y}_t$, after adapting the identity parameters.

$\mathbf{p}_t^{id} \in \mathbb{R}^{n_{id}}$, encoding facial geometry, b) expression parameters $\mathbf{p}_t^{exp} \in \mathbb{R}^{n_{exp}}$, representing facial movements. This enables disentangling facial shape attributes that depend on identity from shape deformations caused by motion. Camera parameters indicate translation, rotation and orthographic scale of the face object within each frame. At the same time, given a single reference image $\mathbf{y}_{ref}$ of the source person, we perform 3DMM fitting and obtain the source's identity shape parameters $\mathbf{p}_{ref}^{id}$, $\mathbf{p}_{ref}^{exp}$ and camera parameters $\mathbf{c}_{ref}$. Then, for each frame t, we compute a 3D facial shape $\mathbf{s}_t = [x_1, y_1, z_1, ..., x_N, y_N, z_N]^\top \in \mathbb{R}^{3N}$, as

$$\mathbf{s}_t = \bar{\mathbf{x}} + \mathbf{U}^{id}\mathbf{p}_{ref}^{id} + \mathbf{U}^{exp}\mathbf{p}_t^{exp}, \tag{1}$$

where $\bar{\mathbf{x}} \in \mathbb{R}^{3N}$ is the mean shape, $\mathbf{U}_{id}$ is the identity orthonormal basis and $\mathbf{U}_{exp}$ is the expression orthonormal basis of LSFM morphable model [5]. By construction, this 3D shape $\mathbf{s}_t$ reflects the facial structure of the source, with the facial expressions of the target. In this way we address the source's identity preservation problem, while we recover very accurate facial expressions from the source, as our 3DMM fitting stage relies on a dense set of 3D points (around 1K), which are regressed from the source frames with RetinaFace [11]. This gives our method an advantage compared to reenactment systems such as [37, 42, 41] that rely on sparse facial landmarks as an intermediate representation of human faces, where source identity preservation problems rise. We render the 3D shape $\mathbf{s}_t$ using the camera parameters $\mathbf{c}_t$, to obtain the 3D face representation $\mathbf{x}_t = \mathcal{R}(\mathbf{s}_t, \mathbf{c}_t)$, $t = 1, \dots, T$, which is an RGB image, as shown in Fig. 2. As a last step, we render $\mathbf{x}_{ref}$, which is the 3D shape reconstructed from the reference image $\mathbf{y}_{ref}$, using $\mathbf{p}_{ref}^{id}$, $\mathbf{p}_{ref}^{exp}$ and $\mathbf{c}_{ref}$.

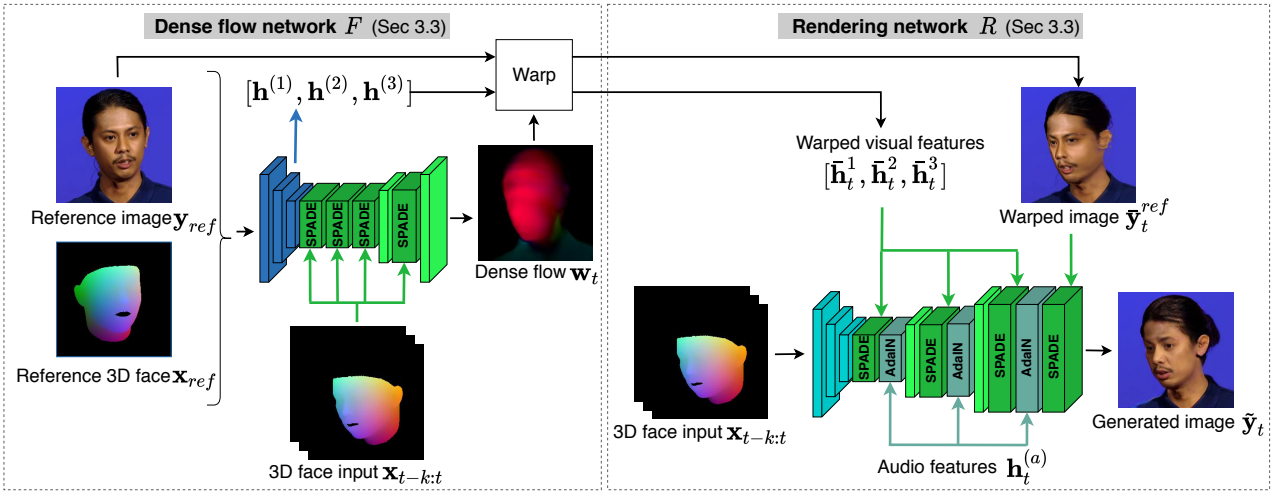


Figure 3: Overview of the proposed Generator G. The dense flow network F computes a flow field for warping the reference image and features, according to the 3D face input. Then, the rendering network R uses this visual information along with the audio features, in order to translate the 3D face input into a photo-realistic image of the source.

Summarising, given a driving video $\mathbf{y}_{1:T}$ and a source image $\mathbf{y}_{ref}$, the data pre-processing pipeline recovers a sequence of images $\mathbf{x}_{1:T}$, which depict the 3D face extracted from the driver and adapted to the facial geometry of the source, as well as the 3D face of the reference image $\mathbf{x}_{ref}$. These face representations are used to condition synthesis.

3.2. Audio features

As opposed to previous one-shot head reenactment systems, our method takes advantage of the driving audio stream and its correlation with facial and mouth movements. We split the audio signal into T parts $\mathbf{a}_{1:T}$, where each part $\mathbf{a}_t$ is aligned and corresponds to frame $\mathbf{y}_t$ of the driving video with length T. Then, we apply our audio feature extraction to a window of $2L$ audio parts $\mathbf{a}_{t-L-1:t+L} = \{\mathbf{a}_{t-L-1}, \ldots, \mathbf{a}_t, \ldots, \mathbf{a}_{t+L}\}$, centred around frame t, to obtain a feature vector $\mathbf{h}_t^{(a)}$, which contains information from the past and future time steps. We employ [12] for the extraction of low level features, such as MFCCs, signal energy and entropy, which yields a feature vector $\mathbf{h}_t^{(a_L)} \in \mathbb{R}^{84}$. Then, we use DeepSpeech [14] for the extraction of character level logits from each part $\mathbf{a}_{t'} \in \mathbf{a}_{t-L-1:t+L}$. This results in $2L$ logits, which after concatenation gives a feature vector $\mathbf{h}_t^{(a_H)} \in \mathbb{R}^{2L \cdot 27}$. Our final audio feature vector is given as $\mathbf{h}_t^{(a)} = [\mathbf{h}_t^{(a_L)^\top}; \mathbf{h}_t^{(a_H)^\top}]^\top \in \mathbb{R}^{300}$, for $L = 4$.

3.3. HeadGAN Framework

We propose a GAN-based head reenactment system, with a Generator driven by two modalities: 1) the *3D face representation* extracted from the driving video and the reference image, 2) the *audio signal* coming from the driver. Given $\mathbf{x}_{t-k:t}$, the driving 3D face representation from frame t, concatenated channel-wise with the 3D faces coming from the past k frames, the reference image $\mathbf{y}_{ref}$ with the corresponding 3D face $\mathbf{x}_{ref}$ and the audio feature vector $\mathbf{h}_t^{(a)}$, our Generator hallucinates a photo-realistic image, given as

$$\tilde{\mathbf{y}}_t = G(\mathbf{x}_{t-k:t}, \mathbf{y}_{ref}, \mathbf{x}_{ref}, \mathbf{h}_t^{(a)}; \theta_G). \quad (2)$$

Conditioning synthesis on the spatio-temporal volume $\mathbf{x}_{t-k:t}$ helps to achieve temporal coherence across frames. The reference image $\mathbf{y}_{ref}$ provides information on the texture and appearance of the source person, while audio features enhance the generative ability of G across the face, and mainly the mouth area. In more detail, the Generator consists of two sub-networks: *a dense flow network* F and a *a rendering network* R. For an overview of the Generator G, please refer to Fig. 3.

Dense flow network F. Our rendering network R relies on high quality visual features that reflect the appearance of the source identity. Nonetheless, we observed that using an encoder to extract such features from the reference image $\mathbf{y}_{ref}$, does not capitalise well on the potential of the rendering network's architecture. It has been proved more meaningful to align the visual feature maps with the desired head pose, which is reflected in the driving 3D face representation $\mathbf{x}_t$, coming from the driving video. With this in mind, we propose a dense flow network, which learns a flow $\mathbf{w}_t$ that can be used to warp visual features. We pass the concatenated reference image and its corresponding 3D face $(\mathbf{y}_{ref}, \mathbf{x}_{ref})$ through an encoder, for the extraction of visual feature maps in three spatial scales $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}$,

which represent the appearance of the source identity. Then, a decoder predicts the flow $\mathbf{w}_t$, guided by the driving 3D face representation $\mathbf{x}_{t-k:t}$, which is injected into F through SPADE blocks [24]. Ideally, when applied on the reference image $\mathbf{y}_{ref}$, this dense flow should yield a warped image of the source person, with the same head pose and expression, as shown in the driving 3D face representation $\mathbf{x}_t$. For the application of flow $\mathbf{w}_t$ on $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$, which have smaller spatial dimensions, we repeatedly down-sample with bilinear interpolation and divide $\mathbf{w}_t$ by two, in order to match it with the spatial dimension of the visual features. Although simplistic, this procedure creates an approximation of the down-sampled flow sufficient for our task. By applying the appropriate flow field on each visual feature map, we obtain the warped visual features $\bar{\mathbf{h}}_t^{(1)}, \bar{\mathbf{h}}_t^{(2)}, \bar{\mathbf{h}}_t^{(3)}$ and the warped reference image $\bar{\mathbf{y}}_t^{ref}$, which depends on the driving head pose at frame t.

Rendering network R. At the core of our generator, the rendering network aims to translate the 3D face representation $\mathbf{x}_{t-k:t}$ into a photo-realistic image $\tilde{\mathbf{y}}_t$ of the source person. This is achieved with the assistance of high quality audio features $\mathbf{h}_t^{(a)}$ and visual feature maps $\bar{\mathbf{h}}_t^{(1)}, \bar{\mathbf{h}}_t^{(2)}, \bar{\mathbf{h}}_t^{(3)}$. First, an encoder receives $\mathbf{x}_{t-k:t}$ as input and applies a sequence of convolutional layers with down-sampling. Then, a decoder consisting of alternating SPADE [24] and AdaIN [16] blocks generates the desired frame $\tilde{\mathbf{y}}_t$. The adaptive instance normalisation layers enable injecting 2D visual features maps and 1D audio features into the rendering network, in order to drive the generative process. As opposed to the original work on SPADE [24], where the conditional input of SPADE layers is the same segmentation map down-sampled to match the spatial size of each layer, we capitalise on visual feature maps of multiple spatial scales $\bar{\mathbf{h}}_t^{(1)}, \bar{\mathbf{h}}_t^{(2)}, \bar{\mathbf{h}}_t^{(3)}, \bar{\mathbf{y}}_t^{ref}$ as conditional input to SPADE blocks. On the contrary, we pass the same audio feature vector $\mathbf{h}_t^{(a)}$ to AdaIn blocks of all spatial scales. The decoder is equipped with PixelShuffle [26] layers for up-sampling, which contributes to the quality of generated samples.

Discriminators D and D_m. Our reenactment system is trained in an adversarial manner, with the assistance of two Discriminators. The image Discriminator receives a synthetic pair $(\mathbf{x}_t, \tilde{\mathbf{y}}_t)$, or a real one $(\mathbf{x}_t, \mathbf{y}_t)$ and learns to distinguish between them. Following the approach of [19], we use a second Discriminator D_m, which focuses on the mouth region. Apart from the real $\mathbf{y}_t^m$ or generated $\tilde{\mathbf{y}}_t^m$ cropped mouth area, this mouth Discriminator is conditioned on the audio feature vector $\mathbf{h}_t^{(a)}$, which is spatially replicated and then concatenated to the cropped images channel-wise. Both these networks follow the architecture of the image Discriminator proposed in [24].

Details on the architecture of *HeadGAN* networks are provided in Appendix B.

4. Experiments

4.1. Implementation details

3D Face Rendering. Given a set of camera parameters $\mathbf{c}$ and a 3D facial shape $\mathbf{s} \in \mathbb{R}^{3N}$ (see Eq. 1), we rasterize the 3D mesh and produce a visibility mask $I \in \mathbb{R}^{H \times W}$ in the image plane. Each spatial location of I stores the index of the corresponding visible triangle on the 3D face seen from this pixel. Then, we use the mean shape $\bar{\mathbf{x}}$ of the 3DMM, in order to find the normalised x-y-z coordinates of the center of each visible triangle. In this way we obtain a 3D face representation $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, where each pixel contains three coordinates, which can be interpreted as colors to give texture the 3D face. This representation carries semantic information, as the color code of each facial part (*e.g.* nose tip) is fixed by construction, regardless of the head pose, enabling the rendering network R to learn the mapping from this 3D face representation, to a photo-realistic image of the source identity.

Dataset and Training. We train and evaluate *HeadGAN* on VoxCeleb [23] dataset, which contains over 100,000 videos of 1,251 identities and 256×256 resolution. We maintain the original train and test split, as they do not intersect with ech other in terms of videos or identities. As a pre-processing step, we compute a 3D face image for each video frame in the dataset and extract per-frame audio feature vectors. During training, we perform self-reenactent, as we randomly sample the reference image from the target video. This allows access to ground truth data, since the source identity now coincides with the target, enabling us to design reconstruction loss terms for optimising the Generator. More details of the objective functions, are provided in Appendix A. For the optimisation of *HeadGAN*, we use the ADAM [18] with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and learning rate $\eta = 0.0002$, both for the Generator and the two Discriminators. We perform five epochs over the training split of VoxCeleb, on two 11GB NVIDIA GeForce RTX2080Ti GPUs, using a batch size of 6.

4.2. Baselines

We compare our approach both quantitatively and qualitatively with the SOTA methods below, under two setups: self-reenactment (source and target identities coincide) and reenactment (source and target identities are different).

X2Face (Wiles *et al*. [39]). This method applies direct image warping. We used the pre-trained model on VoxCeleb, provided by the authors.

Few-shot vid2vid (Wang *et al*. [37]). This is a video-based approach, relying on SPADE [24] architecture and conditioned on facial landmarks. We trained this method on VoxCeleb train split.

Bi-layer Neural Synthesis of Head Avatars (Zakharov *et al*. [41]). This is a fast one-shot head reenactment system,

Figure 4: Qualitative comparison with baselines, under self-reenactment. We illustrate results by increasing difficulty with regards to the distance between the driving and reference head pose, from "easier" (top) to "hard" (bottom).

driven by landmarks. For comparisons, we used the provided model, trained on the larger VoxCeleb2 [8] dataset.

First Order Motion Model (Siarohin *et al*. [28]). This a model-free network that learns to transfer motion from a target video to a reference image. In order to be considered as a reenactment system, the head pose in the first frame of the driving video should match exactly with the one appearing in the reference image. This requirement is satisfied during self-reenactment by choosing the first target frame as the reference image. For comparisons, we use the pre-trained model provided by the authors, trained on VoxCeleb.

We have omitted the comparison with *MarioNETte* [13], as the source code is not publicly available.

4.3. Evaluation metrics

We evaluate *HeadGAN* quantitatively, by performing a numeric comparison with the aforementioned methods. To that end, we conduct a self-reenactment experiment on the test split of [23], and compute the metrics below.

Cosine Similarity (CSIM). Cosine similarity is a widely-used metric, which measures identity preservation in synthetic frames. We use ArcFace [10], an identity recognition network, in order to compute embedding vectors from pairs of ground truth and corresponding generated images. Then, we calculate the cosine similarity between all pairs of embedding vectors in the dataset and report its average value.

Fréchet Inception Distance (FID). We employ FID [15, 25] as a measure of similarity between the dataset of real images and the dataset of images generated by the models. This score provides a useful insight one the photo-realism of synthetic frames.

Fréchet Video Distance (FVD). Given that we handle video data, it is important to evaluate the generative performance of models using a metric for generative models of

video, which takes into account the temporal coherence between frames. To that end, we calculate the FVD score [35] of generated sequences, which has shown to correlate well with qualitative human judgment of generated videos.

Average expression distance (AED). In order to measure the facial expression transferability of models, we perform 3D face reconstruction on the driving as well as the generated images, and measure the average $L1$-distance of the recovered expression parameters, over the entire test set.

4.4. Comparison with Baselines and Results

Quantitative. In Table 1, we report the CSIM, FID, FVD and AED scores achieved by *HeadGAN* and baseline models, on the self-reenactment experiment. As can be observed, our proposed method performs first or second in all metrics. More specifically, *HeadGAN* outperforms all state-of-the-art methods in terms of photo-realism, video quality and temporal coherence, as suggested by FID and FVD scores. We achieve second best performance in identity preservation, behind Siarohin *et al*. [28]. Zakharov *et al*. [41] performs worse than *HeadGAN* on identity preservation but better on the task of facial expression transfer. We attribute this on the fact that it was trained using the larger VoxCeleb2 [8] dataset, something that is supported by our ablation study, in Section 4.5.

Qualitative. In Fig. 4, we visualise examples under self-reenactment. We can see that our method generates faces with crispier details, when compared with the baselines. *HeadGAN* manages to synthesise plausible reenactment results even for extreme pose variations between the source and target. Fig. 6 shows how our method compares with the baselines when reenacting a different identity, as it produces consistently superior results in terms of photo-realism.

4.5. Ablation Study

We conducted an ablation study, in order to assess 1) the role of the dense flow network F, 2) the importance of 3D face conditioning over facial landmarks, 3) the contribution of audio modality in synthesis. As can be seen in Table 1, the full model outperforms all *HeadGAN* variations. In terms of scores, we observe that flow network F is the most essential component of our system. Moreover, AED metric suggests that conditioning on landmarks instead of the 3D face representation leads to inferior performance in expression transfer. This supports our claim that Zakharov *et al*. [41] might perform better in this task, due to VoxCeleb2. In addition, the use of a 3D face instead of facial landmarks solves the identity preservation problem, as seen in Fig. 5b. Lastly, even though the significance of audio input is visible in the metrics, it might be better understood visually, in Fig. 5c, as its effect is more noticeable around the mouth.

Lipreading experiment. We further evaluate the contribution of audio input to our system quantitatively, by employing an external lipreading network to classify synthetic videos. To that end, we chose 25 word classes of BBC dataset [9] and trained a lipreading classifier network [30] on the default training split. After that, we reconstructed the test split of BBC dataset, by performing self reenact-

Method \Metric	CSIM	FID	FVD	AED
Wiles *et al*. [39]	0.620	130.2	697	1.848
Wang *et al*. [37]	0.588	62.8	471	1.872
Zakharov *et al*. [41]	0.625	92.2	394	**1.014**
Siarohin *et al*. [28]	**0.776**	*64.9*	*338*	1.164
HeadGAN	*0.716*	**50.9**	**334**	*1.125*
HeadGAN w/o network F	0.307	63.3	473	1.195
HeadGAN w/o audio input	0.687	55.2	356	1.149
HeadGAN w/ landmarks	0.699	55.7	371	1.162

Table 1: Quantitative results on VoxCeleb test set. For all metrics, except from CSIM, lower is better.

(a) Significance of dense flow network F.

(b) The source identity preservation problem becomes prominent when conditioning on facial landmarks, instead of our 3D face representation.

(c) Contribution of audio modality in mouth region.

Figure 5: The significance of *HeadGAN* components.

Figure 6: Qualitative comparison with baselines when reenacting a source different from the target. All source and target images have not been seen by models during training. Results are presented from "easy" (top) to "hard" (bottom).

ment, using a random frame from the video as reference. We generated data with our full *HeadGAN* model, as well as the variation that does not consider audio input. We report a lipreading accuracy of 97% on real test samples, 82% on samples generated using the the full model and 73% on synthetic data produced by the variation without considering audio input. These results suggest that the audio modality contributes largely on the generation of more plausible lip movements.

5. Conclusion

We presented *HeadGAN*, a novel one-shot method for animating talking heads, driven by 3D facial data, as well as audio features. Compared to state-of-the-art methods, our framework exhibits higher photo-realism, increased consistency in "hard" poses and more accurate lip movements. Future steps include the incorporation of video-modeling components, such as a sequential Generator and temporal Discriminators to further improve the quality of samples.

Appendix A. Objective functions

We train HeadGAN framework, consisting of the Generator G and the two Discriminators D and D_m, using GAN Hinge loss [21]. Therefore, the adversarial loss term for G is given by

$$\mathcal{L}_G^{adv} = -\mathrm{E}_{p_{data}}[D(\mathbf{x}_t, \tilde{\mathbf{y}}_t) + D_m(\mathbf{h}_t^{(a)}, \tilde{\mathbf{y}}_t^m)], \quad (3)$$

where $\mathbf{x}_t$ is the 3D face representation input, $\mathbf{h}_t^{(a)}$ is the input audio feature vector, $\tilde{\mathbf{y}}_t$ is the "fake" frame generated by G and $\tilde{\mathbf{y}}_t^m$ the corresponding cropped mouth area of size 64×64. Given that during training we perform self-reenactment, we have access to the ground truth frame $\mathbf{y}_t$. The image Discriminator D is optimised by minimising the loss

$$\begin{aligned}\mathcal{L}_D^{adv} = -\mathrm{E}_{p_{data}}[&\min(0, -1 + D(\mathbf{x}_t, \mathbf{y}_t) \\ &- \min(0, -1 - D(\mathbf{x}_t, \tilde{\mathbf{y}}_t)].\end{aligned} \quad (4)$$

and the mouth Discriminator D_m using a similar loss

$$\begin{aligned}\mathcal{L}_{D_m}^{adv} = -\mathrm{E}_{p_{data}}[&\min(0, -1 + D_m(\mathbf{x}_t, \mathbf{y}_t^m) \\ &- \min(0, -1 - D_m(\mathbf{x}_t, \tilde{\mathbf{y}}_t^m)].\end{aligned} \quad (5)$$

The generative network G is trained by minimising also a reconstruction loss term between the generated and ground frames, in the image pixel space

$$\mathcal{L}_G^{L1} = \mathrm{E}_{p_{data}}[||\tilde{\mathbf{y}}_t - \mathbf{y}_t||_1], \quad (6)$$

as well as the feature space, using feature maps extracted by a pre-trained VGG network [20]:

$$\mathcal{L}_G^{VGG} = \mathrm{E}_{p_{data}}[\sum_l ||VGG_l(\tilde{\mathbf{y}}_t) - VGG_l(\mathbf{y}_t)||_1]. \quad (7)$$

Similarly with VGG loss, we use the two Discriminators to compute visual features from both real and synthetic frames and compute a feature matching loss $\mathcal{L}_G^{FM}$ that was originally proposed in [40] and has been proven very effective at increasing the photo-realism of generated samples.

In addition, we apply both $L1$ and VGG losses on the warped image $\bar{\mathbf{y}}_t^{ref}$, in order to force the dense flow network F to learn a correct flow from the reference image to the desired head pose, obtaining the loss terms $\mathcal{L}_F^{L1}$ and $\mathcal{L}_F^{VGG}$.

Finally, we improve the temporal consistency of generated frames by imposing a multi-scale $L1$ penalty $\mathcal{L}_F^{Temp}$ on the warped visual features of subsequent frames $(\bar{\mathbf{h}}_{t-1}^{(1)}, \bar{\mathbf{h}}_{t-1}^{(2)}, \bar{\mathbf{h}}_{t-1}^{(3)})$ and $(\bar{\mathbf{h}}_t^{(1)}, \bar{\mathbf{h}}_t^{(2)}, \bar{\mathbf{h}}_t^{(3)})$, given by:

$$\mathcal{L}_F^{Temp} = \sum_{l=1}^{3} ||\bar{\mathbf{h}}_{t-1}^{(l)} - \bar{\mathbf{h}}_t^{(l)}||_1 \quad (8)$$

This loss forces F to use information from the entire temporal volume $\mathbf{x}_{t-k:t}$, which are the driving face representations for k frames (current and past), injected through the normalisation layers of F, for the computation of a smooth flow across time.

To sum up, the overall objective for G is given as:

$$\begin{aligned}\mathcal{L}_G = &\mathcal{L}_G^{adv} + \lambda_{L1}\mathcal{L}_G^{L1} + \lambda_{VGG}\mathcal{L}_G^{VGG} + \lambda_{FM}\mathcal{L}_G^{FM} + \\ &\lambda_{L1}\mathcal{L}_F^{L1} + \lambda_{VGG}\mathcal{L}_F^{VGG} + \lambda_{Temp}\mathcal{L}_F^{Temp},\end{aligned} \quad (9)$$

with $\lambda_{L1} = 50$, $\lambda_{VGG} = \lambda_{FM} = 10$ and $\lambda_{Temp} = 30$. The Discriminators are optimised under their corresponding adversarial loss terms

$$\mathcal{L}_D = \mathcal{L}_D^{adv}, \quad \mathcal{L}_{D_m} = \mathcal{L}_{D_m}^{adv}. \quad (10)$$

Appendix B. Architecture

B.1. Generator G

Dense flow network F (Table 2). The dense flow network consists of an encoding and a decoding part. Its encoder is made up from three convolutional layers, each one with instance normalization units [34] and ReLU activation functions. The last two convolutions are performed with a stride of 2, for down-sampling the input twice. The decoder is equipped with SPADE blocks [24], which are used to "inject" the 3D face representation $\mathbf{x}_{t-k:t}$ (modulation input). Here we down-sample $\mathbf{x}_{t-k:t}$ to match it with the spatial size of each SPADE layer, similarly with the original work [24]. We employ two Pixel Shuffle [26] layers, for up-sampling. Finally, dense flow is calculated with a 7×7 convolutional output layer.

	Block		Output size
	Input		$(256, 256, 6)$
7×7 conv-32	Inst. Norm.	ReLU	$(256, 256, 32)$
3×3 conv-128	Inst. Norm.	ReLU	$(128, 128, 128)$
3×3 conv-512	Inst. Norm.	ReLU	$(64, 64, 512)$
	SPADE Block		$(64, 64, 512)$
	SPADE Block		$(64, 64, 512)$
	SPADE Block		$(64, 64, 512)$
	Pixel Shuffle		$(128, 128, 128)$
	SPADE Block		$(128, 128, 128)$
	Pixel Shuffle		$(256, 256, 32)$
	7×7 conv-2		$(256, 256, 2)$

Table 2: Architecture of dense flow network F.

Rendering network R (Table 3). Our rendering network has an encoder-decoder architecture as well. Its encoder has a similar structure with the encoder of F. The decoder is built from alternating SPADE and AdaIN blocks, which are used to condition synthesis on our multi-scale visual feature maps and audio feature vectors respectively. We use Pixel Shuffle layers for up-sampling, since we noticed

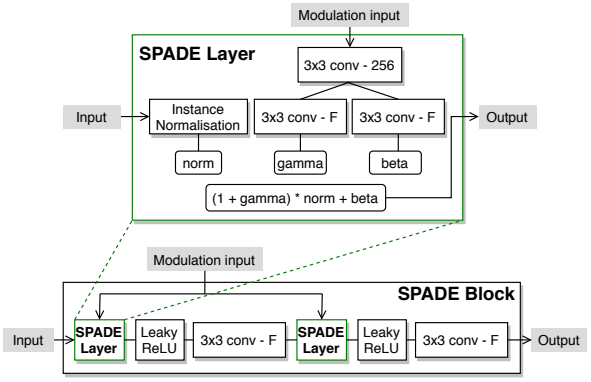
it performs better than simple up-sampling operations (e.g. nearest neighbor, linear, bi-linear). After the last decoding block, a convolutional layer is placed for the computation of the synthetic RGB image.

	Block		Output size
	Input		$(256, 256, 9)$
7×7 conv-32	Inst. Norm.	ReLU	$(256, 256, 32)$
3×3 conv-128	Inst. Norm.	ReLU	$(128, 128, 128)$
3×3 conv-512	Inst. Norm.	ReLU	$(64, 64, 512)$
	SPADE Block		$(64, 64, 512)$
	AdaIN Block		$(64, 64, 512)$
	Pixel Shuffle		$(128, 128, 128)$
	SPADE Block		$(128, 128, 128)$
	AdaIN Block		$(128, 128, 128)$
	Pixel Shuffle		$(256, 256, 32)$
	SPADE Block		$(256, 256, 32)$
	AdaIN Block		$(256, 256, 32)$
	SPADE Block		$(256, 256, 32)$
LReLU	7×7 conv-3	tanh	$(256, 256, 3)$

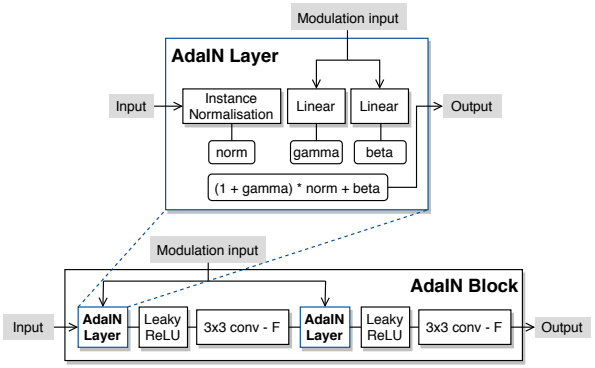
Table 3: Architecture of rendering network R.

B.2. Discriminators D and D_m

Both D and D_m have a similar architecture with the discriminator presented in [24]. We apply Spectral Normalisation [22] to all normalisation layers of the Discriminators.



(a) SPADE Block architecture.



(b) AdaIN Block architecture

Figure 7: Our SPADE and AdaIN blocks are based on the SPADE Resnet blocks proposed in [24], but without a residual component, as we always keep the same number of input channels F at the output, both on SPADE and AdaIN blocks.

References

[1] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F. Cohen. Bringing portraits to life. *ACM Transactions on Graphics (Proceeding of SIGGRAPH Asia 2017)*, 36(6):196, 2017. 2

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2, 3

[3] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *Int. J. Comput. Vision*, 126(2–4):233–254, Apr. 2018. 2, 3

[4] J. Booth, A. Roussos, E. Ververas, E. Antonakos, S. Ploumpis, Y. Panagakis, and S. Zafeiriou. 3d reconstruction of "in-the-wild" faces in images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2638–2652, 2018. 2, 3

[5] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 3

[6] 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. 3

[7] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. You said that? In *British Machine Vision Conference*, 2017. 3

[8] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 2, 6, 7

[9] J. S. Chung and A. Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, 2016. 7

[10] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 6

[11] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[12] Theodoros Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one*, 10(12), 2015. 4

[13] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 1, 2, 6

[14] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Ng. Deepspeech: Scaling up end-to-end speech recognition. 12 2014. 4

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6626–6637. Curran Associates, Inc., 2017. 2, 6

[16] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, 2017. 2, 5

[17] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018. 1, 2, 3

[18] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[19] M. Koujan, M. Doukas, A. Roussos, and S. Zafeiriou. Head2head: Video-based neural head synthesis. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020) (FG)*, pages 319–326, Los Alamitos, CA, USA, may 2020. IEEE Computer Society. 1, 3, 5

[20] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. 9

[21] Jae Hyun Lim and Jong Chul Ye. Geometric gan, 2017. 9

[22] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. 10

[23] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017. 5, 6

[24] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5, 9, 10

[25] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.1.1. 6

[26] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 5, 9

[27] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[28] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Conference on Neural Information Processing Systems (NeurIPS)*, December 2019. 1, 2, 6, 7

[29] Yang Song, Jingwen Zhu, Dawei Li, Andy Wang, and Hairong Qi. Talking face generation by conditional recurrent adversarial network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 919–925. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 2, 3

[30] Themos Stafylakis and Georgios Tzimiropoulos. Combining residual networks with lstms for lipreading. *CoRR*, abs/1703.04105, 2017. 7

[31] Supasorn Suwajanakorn, Steven Seitz, and Ira Kemelmacher. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics*, 36:1–13, 07 2017. 3

[32] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. *ECCV 2020*, 2020. 2, 3

[33] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 1, 2

[34] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. 9

[35] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 2, 7

[36] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *International Journal of Computer Vision*, 10 2019. 3

[37] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 3, 5, 7

[38] 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. 1, 2

[39] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 2, 5, 7

[40] Xiangyu Xu, Deqing Sun, Jinshan Pan, Yujin Zhang, Hanspeter Pfister, and Ming-Hsuan Yang. Learning to super-resolve blurry face and text images. In *Proceedings of the IEEE international conference on computer vision*, pages 251–260, 2017. 9

[41] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *European Conference of Computer vision (ECCV)*, August 2020. 1, 2, 3, 5, 7

[42] E. Zakharov, Aliaksandra Shysheya, Egor Burkov, and V. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9458–9467, 2019. 1, 2, 3