Few-Shot Adversarial Learning of Realistic Neural Talking Head Models

Egor Zakharov[1,2]   Aliaksandra Shysheya[1,2]   Egor Burkov[1,2]   Victor Lempitsky[1,2]

[1]Samsung AI Center, Moscow   [2]Skolkovo Institute of Science and Technology

Figure 1: The results of talking head image synthesis using face landmark tracks extracted from a different video sequence of the same person (on the left), and using face landmarks of a different person (on the right). The **results** are conditioned on the **landmarks** taken from the **target** frame, while the **source** frame is an example from the training set. The talking head models on the left were trained using eight frames, while the models on the right were trained in a one-shot manner.

Abstract

Several recent works have shown how highly realistic human head images can be obtained by training convolutional neural networks to generate them. In order to create a personalized talking head model, these works require training on a large dataset of images of a single person. However, in many practical scenarios, such personalized talking head models need to be learned from a few image views of a person, potentially even a single image. Here, we present a system with such few-shot capability. It performs lengthy meta-learning on a large dataset of videos, and after that is able to frame few- and one-shot learning of neural talking head models of previously unseen people as adversarial training problems with high capacity generators and discriminators. Crucially, the system is able to initialize the parameters of both the generator and the discriminator in a person-specific way, so that training can be based on just a few images and done quickly, despite the need to tune tens of millions of parameters. We show that such an approach is able to learn highly realistic and personalized talking head models of new people and even portrait paintings.

1. Introduction

In this work, we consider the task of creating personalized photorealistic talking head models, i.e. systems that can synthesize plausible video-sequences of speech expressions and mimics of a particular individual. More specifically, we consider the problem of synthesizing photorealistic personalized head images given a set of face landmarks, which drive the animation of the model. Such ability has practical applications for telepresence, including videoconferencing and multi-player games, as well as special effects industry. Synthesizing realistic talking head sequences is known to be hard for two reasons. First, human heads have high photometric, geometric and kinematic complexity. This complexity stems not only from modeling faces (for which a large number of modeling approaches exist) but also from modeling mouth cavity, hair, and garments. The second complicating factor is the acuteness of the human visual system towards even minor mistakes in the appearance modeling of human heads (the so-called *uncanny valley* effect [25]). Such low tolerance to modeling mistakes explains the current prevalence of non-photorealistic cartoon-like avatars in many practically-deployed teleconferencing systems.

To overcome the challenges, several works have proposed to synthesize articulated head sequences by warping a single or multiple static frames. Both classical warping algorithms [4, 30] and warping fields synthesized using machine learning (including deep learning) [11, 31, 42] can be used for such purposes. While warping-based systems can create talking head sequences from as little as a single image, the amount of motion, head rotation, and disocclusion

that they can handle without noticeable artifacts is limited.

Direct (warping-free) synthesis of video frames using adversarially-trained deep convolutional networks (ConvNets) presents the new hope for photorealistic talking heads. Very recently, some remarkably realistic results have been demonstrated by such systems [17, 21, 39]. However, to succeed, such methods have to train large networks, where both generator and discriminator have tens of millions of parameters for each talking head. These systems, therefore, require a several-minutes-long video [21, 39] or a large dataset of photographs [17] as well as hours of GPU training in order to create a new personalized talking head model. While this effort is lower than the one required by systems that construct photo-realistic head models using sophisticated physical and optical modeling [1], it is still excessive for most practical telepresence scenarios, where we want to enable users to create their personalized head models with as little effort as possible.

In this work, we present a system for creating talking head models from a handful of photographs (so-called *few-shot learning*) and with limited training time. In fact, our system can generate a reasonable result based on a single photograph (*one-shot learning*), while adding a few more photographs increases the fidelity of personalization. Similarly to [17, 21, 39], the talking heads created by our model are deep ConvNets that synthesize video frames in a direct manner by a sequence of convolutional operations rather than by warping. The talking heads created by our system can, therefore, handle a large variety of poses that goes beyond the abilities of warping-based systems.

The few-shot learning ability is obtained through extensive pre-training (*meta-learning*) on a large corpus of talking head videos corresponding to different speakers with diverse appearance. In the course of meta-learning, our system simulates few-shot learning tasks and learns to transform landmark positions into realistically-looking personalized photographs, given a small training set of images with this person. After that, a handful of photographs of a new person sets up a new adversarial learning problem with high-capacity generator and discriminator pre-trained via meta-learning. The new adversarial problem converges to the state that generates realistic and personalized images after a few training steps.

In the experiments, we provide comparisons of talking heads created by our system with alternative neural talking head models [17, 42] via quantitative measurements and a user study, where our approach generates images of sufficient realism and personalization fidelity to deceive the study participants. We demonstrate several uses of our talking head models, including video synthesis using landmark tracks extracted from video sequences of the same person, as well as *puppeteering* (video synthesis of a certain person based on the face landmark tracks of a different person).

2. Related work

A huge body of works is devoted to statistical modeling of the appearance of human faces [5], with remarkably good results obtained both with classical techniques [37] and, more recently, with deep learning [23, 26] (to name just a few). While modeling faces is a highly related task to talking head modeling, the two tasks are not identical, as the latter also involves modeling non-face parts such as hair, neck, mouth cavity and often shoulders/upper garment. These non-face parts cannot be handled by some trivial extension of the face modeling methods since they are much less amenable for registration and often have higher variability and higher complexity than the face part. In principle, the results of face modeling [37] or lips modeling [33] can be stitched into an existing head video. Such design, however, does not allow full control over the head rotation in the resulting video and therefore does not result in a fully-fledged talking head system.

The design of our system borrows a lot from the recent progress in generative modeling of images. Thus, our architecture uses adversarial training [12] and, more specifically, the ideas behind conditional discriminators [24], including projection discriminators [34]. Our meta-learning stage uses the adaptive instance normalization mechanism [14], which was shown to be useful in large-scale conditional generation tasks [6, 36]. We also find an idea of content-style decomposition [15] to be extremely useful for separating the texture from the body pose.

The model-agnostic meta-learner (MAML) [10] uses meta-learning to obtain the initial state of an image classifier, from which it can quickly converge to image classifiers of unseen classes, given few training samples. This high-level idea is also utilized by our method, though our implementation of it is rather different. Several works have further proposed to combine adversarial training with meta-learning. Thus, data-augmentation GAN [2], Meta-GAN [45], adversarial meta-learning [43] use adversarially-trained networks to generate additional examples for classes unseen at the meta-learning stage. While these methods are focused on boosting the few-shot classification performance, our method deals with the training of image generation models using similar adversarial objectives. To summarize, we bring the adversarial fine-tuning into the meta-learning framework. The former is applied after we obtain initial state of the generator and the discriminator networks via the meta-learning stage.

Finally, very related to ours are the two recent works on text-to-speech generation [3, 19]. Their setting (few-shot learning of generative models) and some of the components (standalone embedder network, generator fine-tuning) are are also used in our case. Our work differs in the application domain, the use of adversarial learning, its adaptation to the meta-learning process and implementation details.
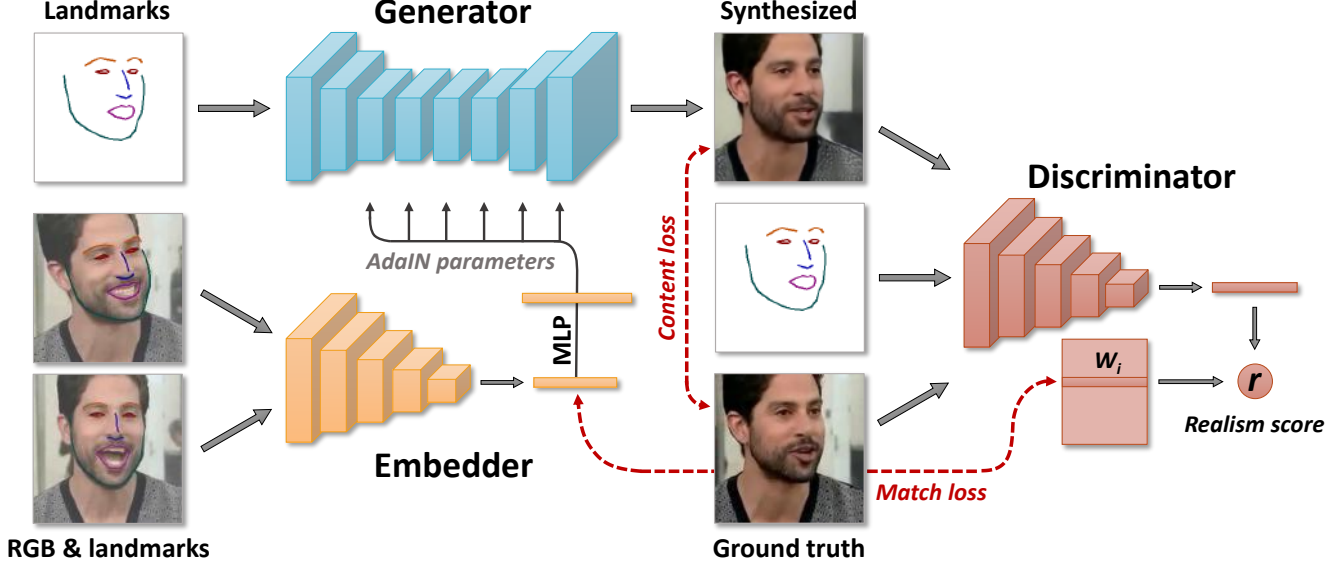
Figure 2: Our meta-learning architecture involves the embedder network that maps head images (with estimated face landmarks) to the embedding vectors, which contain pose-independent information. The generator network maps input face landmarks into output frames through the set of convolutional layers, which are modulated by the embedding vectors via adaptive instance normalization. During meta-learning, we pass sets of frames from the same video through the embedder, average the resulting embeddings and use them to predict adaptive parameters of the generator. Then, we pass the landmarks of a different frame through the generator, comparing the resulting image with the ground truth. Our objective function includes perceptual and adversarial losses, with the latter being implemented via a conditional projection discriminator.

3. Methods

3.1. Architecture and notation

The meta-learning stage of our approach assumes the availability of M video sequences, containing talking heads of different people. We denote with $\mathbf{x}_i$ the i-th video sequence and with $\mathbf{x}_i(t)$ its t-th frame. During the learning process, as well as during test time, we assume the availability of the face landmarks' locations for all frames (we use an off-the-shelf face alignment code [7] to obtain them). The landmarks are rasterized into three-channel images using a predefined set of colors to connect certain landmarks with line segments. We denote with $\mathbf{y}_i(t)$ the resulting *landmark image* computed for $\mathbf{x}_i(t)$.

In the meta-learning stage of our approach, the following three networks are trained (Figure 2):

- The *embedder* $E(\mathbf{x}_i(s), \mathbf{y}_i(s); \phi)$ takes a video frame $\mathbf{x}_i(s)$, an associated landmark image $\mathbf{y}_i(s)$ and maps these inputs into an N-dimensional vector $\hat{\mathbf{e}}_i(s)$. Here, ϕ denotes network parameters that are learned in the meta-learning stage. In general, during meta-learning, we aim to learn ϕ such that the vector $\hat{\mathbf{e}}_i(s)$ contains video-specific information (such as the person's identity) that is invariant to the pose and mimics in a particular frame s. We denote embedding vectors computed by the embedder as $\hat{\mathbf{e}}_i$.

- The *generator* $G(\mathbf{y}_i(t), \hat{\mathbf{e}}_i; \psi, \mathbf{P})$ takes the landmark image $\mathbf{y}_i(t)$ for the video frame not seen by the embedder, the predicted video embedding $\hat{\mathbf{e}}_i$ and outputs a synthesized video frame $\hat{\mathbf{x}}_i(t)$. The generator is trained to maximize the similarity between its outputs and the ground truth frames. All parameters of the generator are split into two sets: the person-generic parameters ψ, and the person-specific parameters $\hat{\psi}_i$. During meta-learning, only ψ are trained directly, while $\hat{\psi}_i$ are predicted from the embedding vector $\hat{\mathbf{e}}_i$ using a trainable projection matrix $\mathbf{P}$: $\hat{\psi}_i = \mathbf{P}\hat{\mathbf{e}}_i$.

- The *discriminator* $D(\mathbf{x}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b)$ takes a video frame $\mathbf{x}_i(t)$, an associated landmark image $\mathbf{y}_i(t)$ and the index of the training sequence i. Here, $\theta, \mathbf{W}, \mathbf{w}_0$ and b denote the learnable parameters associated with the discriminator. The discriminator contains a ConvNet part $V(\mathbf{x}_i(t), \mathbf{y}_i(t); \theta)$ that maps the input frame and the landmark image into an N-dimensional vector. The discriminator predicts a single scalar (realism score) r, that indicates whether the input frame $\mathbf{x}_i(t)$ is a real frame of the i-th video sequence and whether it matches the input pose $\mathbf{y}_i(t)$, based on the output of its ConvNet part and the parameters $\mathbf{W}, \mathbf{w}_0, b$.

3.2. Meta-learning stage

During the meta-learning stage of our approach, the parameters of all three networks are trained in an adversarial

fashion. It is done by simulating episodes of K-shot learning ($K = 8$ in our experiments). In each episode, we randomly draw a training video sequence i and a single frame t from that sequence. In addition to t, we randomly draw additional K frames $s_1, s_2, \ldots, s_K$ from the same sequence. We then compute the estimate $\hat{\mathbf{e}}_i$ of the i-th video embedding by simply averaging the embeddings $\hat{\mathbf{e}}_i(s_k)$ predicted for these additional frames:

$$\hat{\mathbf{e}}_i = \frac{1}{K} \sum_{k=1}^{K} E\left(\mathbf{x}_i(s_k), \mathbf{y}_i(s_k); \phi\right). \tag{1}$$

A reconstruction $\hat{\mathbf{x}}_i(t)$ of the t-th frame, based on the estimated embedding $\hat{\mathbf{e}}_i$, is then computed:

$$\hat{\mathbf{x}}_i(t) = G\left(\mathbf{y}_i(t), \hat{\mathbf{e}}_i; \psi, \mathbf{P}\right). \tag{2}$$

The parameters of the embedder and the generator are then optimized to minimize the following objective that comprises the content term, the adversarial term, and the embedding match term:

$$\mathcal{L}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \mathcal{L}_{\mathrm{CNT}}(\phi, \psi, \mathbf{P}) + \tag{3}$$
$$\mathcal{L}_{\mathrm{ADV}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) + \mathcal{L}_{\mathrm{MCH}}(\phi, \mathbf{W}).$$

In (3), the content loss term $\mathcal{L}_{\mathrm{CNT}}$ measures the distance between the ground truth image $\mathbf{x}_i(t)$ and the reconstruction $\hat{\mathbf{x}}_i(t)$ using the perceptual similarity measure [20], corresponding to VGG19 [32] network trained for ILSVRC classification and VGGFace [28] network trained for face verification. The loss is calculated as the weighted sum of L_1 losses between the features of these networks.

The adversarial term in (3) corresponds to the realism score computed by the discriminator, which needs to be maximized, and a feature matching term [40], which essentially is a perceptual similarity measure, computed using discriminator (it helps with the stability of the training):

$$\mathcal{L}_{\mathrm{ADV}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \tag{4}$$
$$-D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b) + \mathcal{L}_{\mathrm{FM}}.$$

Following the projection discriminator idea [34], the columns of the matrix $\mathbf{W}$ contain the embeddings that correspond to individual videos. The discriminator first maps its inputs to an N-dimensional vector $V(\mathbf{x}_i(t), \mathbf{y}_i(t); \theta)$ and then computes the realism score as:

$$D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b) = \tag{5}$$
$$V\left(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t); \theta\right)^T (\mathbf{W}_i + \mathbf{w}_0) + b,$$

where $\mathbf{W}_i$ denotes the i-th column of the matrix $\mathbf{W}$. At the same time, $\mathbf{w}_0$ and b do not depend on the video index, so these terms correspond to the general realism of $\hat{\mathbf{x}}_i(t)$ and its compatibility with the landmark image $\mathbf{y}_i(t)$.

Thus, there are two kinds of video embeddings in our system: the ones computed by the embedder, and the ones that correspond to the columns of the matrix $\mathbf{W}$ in the discriminator. The match term $\mathcal{L}_{\mathrm{MCH}}(\phi, \mathbf{W})$ in (3) encourages the similarity of the two types of embeddings by penalizing the L_1-difference between $E\left(\mathbf{x}_i(s_k), \mathbf{y}_i(s_k); \phi\right)$ and $\mathbf{W}_i$.

As we update the parameters ϕ of the embedder and the parameters ψ of the generator, we also update the parameters $\theta, \mathbf{W}, \mathbf{w}_0, b$ of the discriminator. The update is driven by the minimization of the following hinge loss, which encourages the increase of the realism score on real images $\mathbf{x}_i(t)$ and its decrease on synthesized images $\hat{\mathbf{x}}_i(t)$:

$$\mathcal{L}_{\mathrm{DSC}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \tag{6}$$
$$\max(0, 1 + D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i; \phi, \psi, \theta, \mathbf{W}, \mathbf{w}_0, b)) +$$
$$\max(0, 1 - D(\mathbf{x}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b)).$$

The objective (6) thus compares the realism of the fake example $\hat{\mathbf{x}}_i(t)$ and the real example $\mathbf{x}_i(t)$ and then updates the discriminator parameters to push these scores below -1 and above $+1$ respectively. The training proceeds by alternating updates of the embedder and the generator that minimize the losses $\mathcal{L}_{\mathrm{CNT}}, \mathcal{L}_{\mathrm{ADV}}$ and $\mathcal{L}_{\mathrm{MCH}}$ with the updates of the discriminator that minimize the loss $\mathcal{L}_{\mathrm{DSC}}$.

3.3. Few-shot learning by fine-tuning

Once the meta-learning has converged, our system can learn to synthesize talking head sequences for a new person, unseen during meta-learning stage. As before, the synthesis is conditioned on the landmark images. The system is learned in a few-shot way, assuming that T training images $\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(T)$ (e.g. T frames of the same video) are given and that $\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(T)$ are the corresponding landmark images. Note that the number of frames T needs not to be equal to K used in the meta-learning stage.

Naturally, we can use the meta-learned embedder to estimate the embedding for the new talking head sequence:

$$\hat{\mathbf{e}}_{\mathrm{NEW}} = \frac{1}{T} \sum_{t=1}^{T} E(\mathbf{x}(t), \mathbf{y}(t); \phi), \tag{7}$$

reusing the parameters ϕ estimated in the meta-learning stage. A straightforward way to generate new frames, corresponding to new landmark images, is then to apply the generator using the estimated embedding $\hat{\mathbf{e}}_{\mathrm{NEW}}$ and the meta-learned parameters ψ, as well as projection matrix $\mathbf{P}$. By doing so, we have found out that the generated images are plausible and realistic, however, there often is a considerable identity gap that is not acceptable for most applications aiming for high personalization degree.

This identity gap can often be bridged via the *fine-tuning stage*. The fine-tuning process can be seen as a simplified version of meta-learning with a single video sequence and a

smaller number of frames. The fine-tuning process involves the following components:

- The generator $G(\mathbf{y}(t), \hat{\mathbf{e}}_{\text{NEW}}; \psi, \mathbf{P})$ is now replaced with $G'(\mathbf{y}(t); \psi, \psi')$. As before, it takes the landmark image $\mathbf{y}(t)$ and outputs the synthesized frame $\hat{\mathbf{x}}(t)$. Importantly, the person-specific generator parameters, which we now denote with ψ', are now directly optimized alongside the person-generic parameters ψ. We still use the computed embeddings $\hat{\mathbf{e}}_{\text{NEW}}$ and the projection matrix $\mathbf{P}$ estimated at the meta-learning stage to initialize ψ', i.e. we start with $\psi' = \mathbf{P}\hat{\mathbf{e}}_{\text{NEW}}$.
- The discriminator $D'(\mathbf{x}(t), \mathbf{y}(t); \theta, \mathbf{w}', b)$, as before, computes the realism score. Parameters θ of its ConvNet part $V(\mathbf{x}(t), \mathbf{y}(t); \theta)$ and bias b are initialized to the result of the meta-learning stage. The initialization of $\mathbf{w}'$ is discussed below.

During fine-tuning, the realism score of the discriminator is obtained in a similar way to the meta-learning stage:

$$D'(\hat{\mathbf{x}}(t), \mathbf{y}(t); \theta, \mathbf{w}', b) = \qquad (8)$$
$$V(\hat{\mathbf{x}}(t), \mathbf{y}(t); \theta)^T \mathbf{w}' + b.$$

As can be seen from the comparison of expressions (5) and (8), the role of the vector $\mathbf{w}'$ in the fine-tuning stage is the same as the role of the vector $\mathbf{W}_i + \mathbf{w}_0$ in the meta-learning stage. For the intiailization, we do not have access to the analog of $\mathbf{W}_i$ for the new personality (since this person is not in the meta-learning dataset). However, the match term $\mathcal{L}_{\text{MCH}}$ in the meta-learning process ensures the similarity between the discriminator video-embeddings and the vectors computed by the embedder. Hence, we can initialize $\mathbf{w}'$ to the sum of $\mathbf{w}_0$ and $\hat{\mathbf{e}}_{\text{NEW}}$.

Once the new learning problem is set up, the loss functions of the fine-tuning stage follow directly from the meta-learning variants. Thus, the generator parameters ψ and ψ' are optimized to minimize the simplified objective:

$$\mathcal{L}'(\psi, \psi', \theta, \mathbf{w}', b) = \qquad (9)$$
$$\mathcal{L}'_{\text{CNT}}(\psi, \psi') + \mathcal{L}'_{\text{ADV}}(\psi, \psi', \theta, \mathbf{w}', b),$$

where $t \in \{1 \dots T\}$ is the number of the training example. The discriminator parameters $\theta, \mathbf{w}_{\text{NEW}}, b$ are optimized by minimizing the same hinge loss as in (6):

$$\mathcal{L}'_{\text{DSC}}(\psi, \psi', \theta, \mathbf{w}', b) = \qquad (10)$$
$$\max(0, 1 + D(\hat{\mathbf{x}}(t), \mathbf{y}(t); \psi, \psi', \theta, \mathbf{w}', b)) +$$
$$\max(0, 1 - D(\mathbf{x}(t), \mathbf{y}(t); \theta, \mathbf{w}', b)).$$

In most situations, the fine-tuned generator provides a much better fit of the training sequence. The initialization of all parameters via the meta-learning stage is also crucial. As we show in the experiments, such initialization injects a strong realistic talking head prior, which allows our model to extrapolate and predict realistic images for poses with varying head poses and facial expressions.

3.4. Implementation details

We base our generator network $G(\mathbf{y}_i(t), \hat{\mathbf{e}}_i; \psi, \mathbf{P})$ on the image-to-image translation architecture proposed by Johnson et. al. [20], but replace downsampling and upsampling layers with residual blocks similarly to [6] (with batch normalization [16] replaced by instance normalization [38]). The person-specific parameters $\hat{\psi}_i$ serve as the affine coefficients of instance normalization layers, following the adaptive instance normalization technique proposed in [14], though we still use regular (non-adaptive) instance normalization layers in the downsampling blocks that encode landmark images $\mathbf{y}_i(t)$.

For the embedder $E(\mathbf{x}_i(s), \mathbf{y}_i(s); \phi)$ and the convolutional part of the discriminator $V(\mathbf{x}_i(t), \mathbf{y}_i(t); \theta)$, we use similar networks, which consist of residual downsampling blocks (same as the ones used in the generator, but without normalization layers). The discriminator network, compared to the embedder, has an additional residual block at the end, which operates at 4×4 spatial resolution. To obtain the vectorized outputs in both networks, we perform global sum pooling over spatial dimensions followed by ReLU.

We use spectral normalization [35] for all convolutional and fully connected layers in all the networks. We also use self-attention blocks, following [6] and [44]. They are inserted at 32×32 spatial resolution in all downsampling parts of the networks and at 64×64 resolution in the upsampling part of the generator.

For the calculation of $\mathcal{L}_{\text{CNT}}$, we evaluate L_1 loss between activations of `Conv1,6,11,20,29` VGG19 layers and `Conv1,6,11,18,25` VGGFace layers for real and fake images. We sum these losses with the weights equal to $1.5 \cdot 10^{-1}$ for VGG19 and $2.5 \cdot 10^{-2}$ for VGGFace terms. We use Caffe [18] trained versions for both of these networks. For $\mathcal{L}_{\text{FM}}$, we use activations after each residual block of the discriminator network and the weights equal to 10. Finally, for $\mathcal{L}_{\text{MCH}}$ we also set the weight to 10.

We set the minimum number of channels in convolutional layers to 64 and the maximum number of channels as well as the size N of the embedding vectors to 512. In total, the embedder has 15 million parameters, the generator has 38 million parameters. The convolutional part of the discriminator has 20 million parameters. The networks are optimized using Adam [22]. We set the learning rate of the embedder and the generator networks to 5×10^{-5} and to 2×10^{-4} for the discriminator, doing two update steps for the latter per one of the former, following [44].

4. Experiments

Two datasets with talking head videos are used for quantitative and qualitative evaluation: VoxCeleb1 [27] (256p videos at 1 fps) and VoxCeleb2 [8] (224p videos at 25 fps), with the latter having approximately 10 times more videos

Method (T)	FID↓	SSIM↑	CSIM↑	USER↓
	VoxCeleb1			
X2Face (1)	45.8	**0.68**	**0.16**	0.82
Pix2pixHD (1)	**42.7**	0.56	0.09	0.82
Ours (1)	43.0	0.67	0.15	**0.62**
X2Face (8)	51.5	**0.73**	**0.17**	0.83
Pix2pixHD (8)	**35.1**	0.64	0.12	0.79
Ours (8)	38.0	0.71	**0.17**	**0.62**
X2Face (32)	56.5	**0.75**	0.18	0.85
Pix2pixHD (32)	**24.0**	0.70	0.16	0.71
Ours (32)	29.5	0.74	**0.19**	**0.61**
	VoxCeleb2			
Ours-FF (1)	**46.1**	0.61	**0.42**	**0.43**
Ours-FT (1)	48.5	**0.64**	0.35	0.46
Ours-FF (8)	42.2	0.64	**0.47**	0.40
Ours-FT (8)	**42.2**	**0.68**	0.42	**0.39**
Ours-FF (32)	40.4	0.65	**0.48**	0.38
Ours-FT (32)	**30.6**	**0.72**	0.45	**0.33**

Table 1: Quantitative comparison of methods on different datasets with multiple few-shot learning settings. Please refer to the text for more details and discussion.

than the former. VoxCeleb1 is used for comparison with baselines and ablation studies, while by using VoxCeleb2 we show the full potential of our approach.

Metrics. For the quantitative comparisons, we fine-tune all models on few-shot learning sets of size T for a person not seen during meta-learning (or pretraining) stage. After the few-shot learning, the evaluation is performed on the hold-out part of the same sequence (so-called *self-reenactment* scenario). For the evaluation, we uniformly sampled 50 videos from VoxCeleb test sets and 32 hold-out frames for each of these videos (the fine-tuning and the hold-out parts do not overlap).

We use multiple comparison metrics to evaluate photo-realism and identity preservation of generated images. Namely, we use Frechet-inception distance (FID) [13], mostly measuring perceptual realism, structured similarity (SSIM) [41], measuring low-level similarity to the ground truth images, and cosine similarity (CSIM) between embedding vectors of the state-of-the-art face recognition network [9] for measuring identity mismatch (note that this network has quite different architecture from VGGFace used within content loss calculation during training).

We also perform a user study in order to evaluate perceptual similarity and realism of the results as seen by the human respondents. We show people the triplets of images of the same person taken from three different video sequences. Two of these images are real and one is fake, produced by one of the methods, which are being compared. We ask the user to find the fake image given that all of these images are

of the same person. This evaluates both photo-realism and identity preservation because the user can infer the identity from the two real images (and spot an identity mismatch even if the generated image is perfectly realistic). We use the user accuracy (success rate) as our metric. The lower bound here is the accuracy of one third (when users cannot spot fakes based on non-realism or identity mismatch and have to guess randomly). Generally, we believe that this user-driven metric (USER) provides a much better idea of the quality of the methods compared to FID, SSIM, or CSIM.

Methods. On the VoxCeleb1 dataset we compare our model against two other systems: X2Face [42] and Pix2pixHD [40]. For X2Face, we have used the model, as well as pretrained weights, provided by the authors (in the original paper it was also trained and evaluated on the Vox-Celeb1 dataset). For Pix2pixHD, we pretrained the model from scratch on the whole dataset for the same amount of iterations as our system without any changes to thetraining pipeline proposed by the authors. We picked X2Face as a strong baseline for warping-based methods and Pix2pixHD for direct synthesis methods.

In our comparison, we evaluate the models in several scenarios by varying the number of frames T used in few-shot learning. X2Face, as a feed-forward method, is simply initialized via the training frames, while Pix2pixHD and our model are being additionally fine-tuned for 40 epochs on the few-shot set. Notably, in the comparison, X2Face uses dense correspondence field, computed on the ground truth image, to synthesize the generated one, while our method and Pix2pixHD use very sparse landmark information, which arguably gives X2Face an unfair advantage.

Comparison results. We perform comparison with baselines in three different setups, with 1, 8 and 32 frames in the fine-tuning set. Test set, as mentioned before, consists of 32 hold-out frames for each of the 50 test video sequences. Moreover, for each test frame we sample two frames at random from the other video sequences with the same person. These frames are used in triplets alongside with fake frames for user-study.

As we can see in Table 1-Top, baselines consistently outperform our method on the two of our similarity metrics. We argue that this is intrinsic to the methods themselves: X2Face uses L_2 loss during optimization [42], which leads to a good SSIM score. On the other hand, Pix2pixHD maximizes only perceptual metric, without identity preservation loss, leading to minimization of FID, but has bigger identity mismatch, as seen from the CSIM column. Moreover, these metrics do not correlate well with human perception, since both of these methods produce uncanny valley artifacts, as can be seen from qualitative comparison Figure 3 and the

Figure 3: Comparison on the VoxCeleb1 dataset. For each of the compared methods, we perform one- and few-shot learning on a video of a person not seen during meta-learning or pretraining. We set the number of training frames equal to **T** (the leftmost column). One of the training frames is shown in the **source** column. Next columns show **ground truth** image, taken from the test part of the video sequence, and the generated results of the compared methods.

user study results. Cosine similarity, on the other hand, better correlates with visual quality, but still favours blurry, less realistic images, and that can also be seen by comparing Table 1-Top with the results presented in Figure 3.

While the comparison in terms of the objective metrics is inconclusive, the user study (that included 4800 triplets, each shown to 5 users) clearly reveals the much higher realism and personalization degree achieved by our method.

We have also carried out the ablation study of our system and the comparison of the few-shot learning timings. Both are provided in the Supplementary material.

Large-scale results. We then scale up the available data and train our method on a larger VoxCeleb2 dataset. Here, we train two variants of our method. FF (feed-forward) variant is trained for 150 epochs without the embedding matching loss $\mathcal{L}_{\text{MCH}}$ and, therefore, we only use it without fine-tuning (by simply predicting adaptive parameters ψ' via the projection of the embedding $\hat{e}_{\text{NEW}}$). The FT variant is trained for half as much (75 epochs) but with $\mathcal{L}_{\text{MCH}}$, which allows fine-tuning. We run the evaluation for both of these models since they allow to trade off few-shot learning speed versus the results quality. Both of them achieve con-

siderably higher scores, compared to smaller-scale models trained on VoxCeleb1. Notably, the FT model reaches the lower bound of 0.33 for the user study accuracy in $T = 32$ setting, which is a perfect score. We present results for both of these models in Figure 4 and more results (including results, where animation is driven by landmarks from a different video of the same person) are given in the supplementary material and in Figure 1.

Generally, judging by the results of comparisons (Table 1-Bottom) and the visual assessment, the FF model performs better for low-shot learning (e.g. one-shot), while the FT model achieves higher quality for bigger T via adversarial fine-tuning.

Puppeteering results. Finally, we show the results for the puppeteering of photographs and paintings. For that, we evaluate the model, trained in one-shot setting, on poses from test videos of the VoxCeleb2 dataset. We rank these videos using CSIM metric, calculated between the original image and the generated one. This allows us to find persons with similar geometry of the landmarks and use them for the puppeteering. The results can be seen in Figure 5 as well as in Figure 1.

7

Figure 4: Results for our best models on the VoxCeleb2 dataset. The number of training frames is, again, equal to **T** (the leftmost column) and the example training frame in shown in the **source** column. Next columns show **ground truth** image and the results for **Ours-FF** feed-forward model, **Ours-FT** model **before** and **after fine-tuning**. While the feed-forward variant allows fast (real-time) few-shot learning of new avatars, fine-tuning ultimately provides better realism and fidelity.
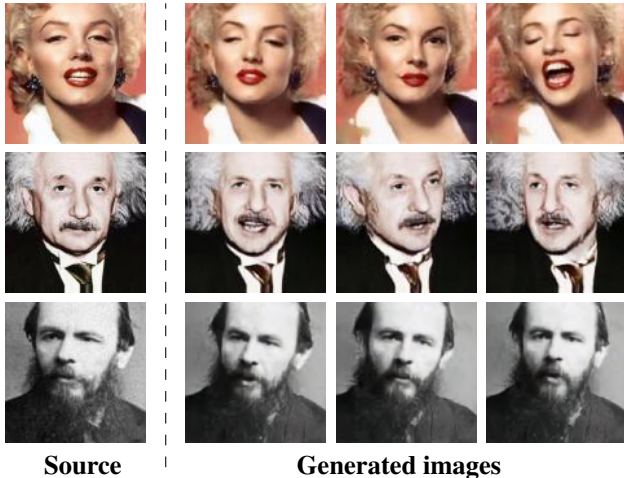

Figure 5: Bringing still photographs to life. We show the puppeteering results for one-shot models learned from photographs in the **source** column. Driving poses were taken from the VoxCeleb2 dataset. Digital zoom recommended.

5. Conclusion

We have presented a framework for meta-learning of adversarial generative models, which is able to train highly-realistic virtual talking heads in the form of deep generator networks. Crucially, only a handful of photographs (as little as one) is needed to create a new model, whereas the model trained on 32 images achieves perfect realism and personalization score in our user study (for 224p static images).

Currently, the key limitations of our method are the mimics representation (in particular, the current set of landmarks does not represent the gaze in any way) and the lack of landmark adaptation. Using landmarks from a different person leads to a noticeable personality mismatch. So, if one wants to create "fake" puppeteering videos without such mismatch, some landmark adaptation is needed. We note, however, that many applications do not require puppeteering a different person and instead only need the ability to drive one's own talking head. For such scenario, our approach already provides a high-realism solution.

References

[1] Oleg Alexander, Mike Rogers, William Lambeth, Jen-Yuan Chiang, Wan-Chun Ma, Chuan-Chang Wang, and Paul Debevec. The Digital Emily project: Achieving a photorealistic digital actor. *IEEE Computer Graphics and Applications*, 30(4):20–31, 2010. 2

[2] Antreas Antoniou, Amos J. Storkey, and Harrison Edwards. Augmenting image classifiers using data augmentation generative adversarial networks. In *Artificial Neural Networks and Machine Learning - ICANN*, pages 594–603, 2018. 2

[3] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In *Proc. NIPS*, pages 10040–10050, 2018. 2

[4] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F Cohen. Bringing portraits to life. *ACM Transactions on Graphics (TOG)*, 36(6):196, 2017. 1, 14

[5] Volker Blanz, Thomas Vetter, et al. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH*, volume 99, pages 187–194, 1999. 2

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 2, 5, 12

[7] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230, 000 3d facial landmarks). In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1021–1030, 2017. 3

[8] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 6

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

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. ICML*, pages 1126–1135, 2017. 2

[11] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *European Conference on Computer Vision*, pages 311–326. Springer, 2016. 1

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[13] 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 30*, pages 6626–6637. Curran Associates, Inc., 2017. 6

[14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. ICCV*, 2017. 2, 5

[15] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015. 5

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, pages 5967–5976, 2017. 2

[18] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[19] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Proc. NIPS*, pages 4485–4495, 2018. 2

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, pages 694–711, 2016. 4, 5

[21] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *arXiv preprint arXiv:1805.11714*, 2018. 2

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[23] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)*, 37(4):68, 2018. 2

[24] Simon Osindero Mehdi Mirza. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014. 2

[25] Masahiro Mori. The uncanny valley. *Energy*, 7(4):33–35, 1970. 1

[26] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, Hao Li, Richard Roberts, et al. paGAN: real-time avatars using dynamic textures. In *SIGGRAPH Asia 2018 Technical Papers*, page 258. ACM, 2018. 2

[27] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017. 5

[28] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. BMVC*, 2015. 4

[29] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2018. 14

[30] Steven M Seitz and Charles R Dyer. View morphing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 21–30. ACM, 1996. 1

9

[31] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 4

[33] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing Obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017. 2

[34] Masanori Koyama Takeru Miyato. cgans with projection discriminator. *arXiv:1802.05637*, 2018. 2, 4

[35] Masanori Koyama Yuichi Yoshida Takeru Miyato, Toshiki Kataoka. Spectral normalization for generative adversarial networks. *arXiv:1802.05957*, 2018. 5

[36] Timo Aila Tero Karras, Samuli Laine. A style-based generator architecture for generative adversarial networks. *arXiv:1812.04948*, 2018. 2

[37] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of RGB videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 2, 12

[38] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. 5, 12

[39] 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. 2

[40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 4, 6

[41] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, Apr. 2004. 6

[42] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2, 6

[43] Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. Adversarial meta-learning. *CoRR*, abs/1806.03316, 2018. 2

[44] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. 5, 12

[45] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *NeurIPS*, pages 2371–2380, 2018. 2

A. Supplementary material

In the supplementary material, we provide additional qualitative results as well as an ablation study and a time comparison between our method and the baselines for both inference and training.

A.1. Time comparison results.

In Table 2, we provide a comparison of timings for the three methods. Additionally, we included the feed-forward variant of our method in the comparison, which was trained only for the VoxCeleb2 dataset. The comparison was carried out on a single NVIDIA P40 GPU. For Pix2pixHD and our method, few-shot learning was done via fine-tuning for 40 epochs on the training set of size T. For T larger than 1, we trained the models on batches of 8 images. Each measurement was averaged over 100 iterations.

We see that, given enough training data, our method in feed-forward variant can outpace all other methods by a large margin in terms of few-shot training time, while keeping personalization fidelity and realism of the outputs on quite a high level (as can be seen in Figure 4). But in order to achieve the best results in terms of quality, fine-tuning has to be performed, which takes approximately four and a half minutes on the P40 GPU for 32 training images. The number of epochs and, hence, the fine-tuning speed can be optimized further on a case by case basis or via the introduction of a training scheduler, which we did not perform.

On the other hand, inference speed for our method is comparable or slower than other methods, which is caused by a large number of parameters we need to encode the prior knowledge about talking heads. Though, this figure can be drastically improved via the usage of more modern GPUs (on an NVIDIA 2080 Ti, the inference time can be decreased down to 13ms per frame, which is enough for most real-time applications).

A.2. Ablation study

In this section, we evaluate the contributions related to the losses we use in the training of our model, as well as motivate the training procedure. We have already shown in Figure 4 the effect that the fine-tuning has on the quality of the results, so we do not evaluate it here. Instead, we focus on the details of fine-tuning.

The first question we asked was about the importance of person-specific parameters initialization via the embedder. We tried different types of random initialization for both the embedding vector $\hat{e}_{NEW}$ and the adaptive parameters $\hat{\psi}$ of the generator, but these experiments did not yield any plausible images after the fine-tuning. Hence we realized that the person-specific initialization of the generator provided by the embedder is important for convergence of the fine-tuning problem.

Method (T)	Time, s
Few-shot learning	
X2Face (1)	0.236
Pix2pixHD (1)	33.92
Ours (1)	43.84
Ours-FF (1)	**0.061**
X2Face (8)	1.176
Pix2pixHD (8)	52.40
Ours (8)	85.48
Ours-FF (8)	**0.138**
X2Face (32)	7.542
Pix2pixHD (32)	122.6
Ours (32)	258.0
Ours-FF (32)	**0.221**
Inference	
X2Face	0.110
Pix2pixHD	**0.034**
Ours	0.139

Table 2: Quantitative comparison of few-shot learning and inference timings for the three models.

Then, we evaluated the contribution of the person-specific initialization of the discriminator. We remove $\mathcal{L}_{MCH}$ term from the objective and perform meta-learning. The use of multiple training frames in few-shot learning problems, like in our final method, leads to optimization instabilities, so we used a one-shot meta-learning configuration, which turned out to be stable. After meta-learning, we randomly initialize the person-specific vector $\mathbf{W}_i$ of the discriminator. The results can be seen in Figure 7. We notice that the results for random initialization are plausible but introduce a noticeable gap in terms of realism and personalization fidelity. We, therefore, came to the conclusion that person-specific initialization of the discriminator also contributes to the quality of the results, albeit in a lesser way than the initialization of the generator does.

Finally, we evaluate the contribution of adversarial term $\mathcal{L}'_{ADV}$ during the fine-tuning. We, therefore, remove it from the fine-tuning objective and compare the results to our best model (see Figure 7). While the difference between these variants is quite subtle, we note that adversarial fine-tuning leads to crisper images that better match ground truth both in terms of pose and image details. The close-up images in Figure 8 were chosen in order to highlight these differences.

A.3. Additional qualitative results

More comparisons with other methods are available in Figure 9, Figure 10, Figure 6. More puppeteering results for one-shot learned portraits and photographs are presented in Figure 11. We also show the results for talking heads learned from selfies in Figure 13. Additional comparisons between the methods are provided in the rest of the figures.
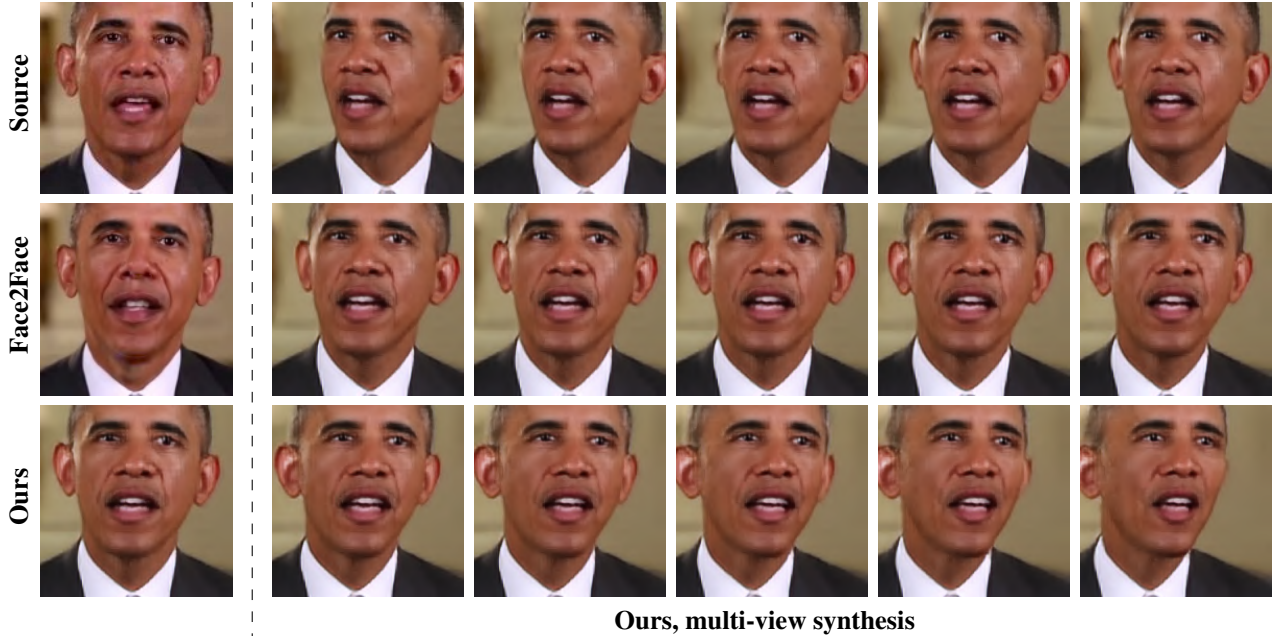
Figure 6: Comparison with Thies et al.[37]. We used 32 frames for the fine-tuning, while 1100 frames were used to train the Face2Face model. Note that the output resolution of our model is constrained by the training dataset. Also, our model is able to synthesize a naturally looking frame from different viewpoints for a fixed pose (given 3D face landmarks), which is a limitation of the Face2Face system.

A.4. Training and architecture details

As stated in the paper, we used the architecture similar to the one in [6]. The convolutional parts of the embedder and the discriminator are the same networks with 6 residual downsampling blocks, each performing downsampling by a factor of 2. The inputs of these convolutional networks are RGB images concatenated with the landmark images, in total there are 6 input channels. The initial number of channels is 64, increased by a factor of two in each block, up to a maximum of 512. The blocks are pre-activated residual blocks with no normalization, as described in the paper [6]. The first block is a regular residual block with activation function not being applied in the end. Each skip connection has a linear layer inside if the spatial resolution is being changed. Self-attention [44] blocks are inserted after three downsampling blocks. Downsampling is performed via average pooling. Then, after applying ReLU activation function to the output tensor, we perform sum-pooling over spatial dimensions.

For the embedder, the resulting vectorized embeddings for each training image are stored (in order to apply $\mathcal{L}_{\text{MCH}}$ element-wise), and the averaged embeddings are fed into the generator. For the discriminator, the resulting vector is used to calculate the realism score.

The generator consists of three parts: 4 residual downsampling blocks (with self-attention inserted before the last

block), 4 blocks operating at bottleneck resolution and 4 upsampling blocks (self-attention is inserted after 2 upsampling blocks). Upsampling is performed in the end of the block, following [6]. The number of channels in bottleneck layers is 512. Downsampling blocks are normalized via instance normalization [38], while bottleneck and upsampling blocks are normalized via adaptive instance normalization. A single linear layer is used to map an embedding vector to all adaptive parameters. After the last upsampling block, we insert a final adaptive normalization layer, followed by a ReLU and a convolution. The output is then mapped into $[-1, 1]$ via Tanh.

The training was carried out on 8 NVIDIA P40 GPUs, with batch size 48 via simultaneous gradient descend, with 2 updates of the discriminator per 1 of the generator. In our experiments, we used PyTorch distributed module and have performed reduction of the gradients across the GPUs only for the generator and the embedder.
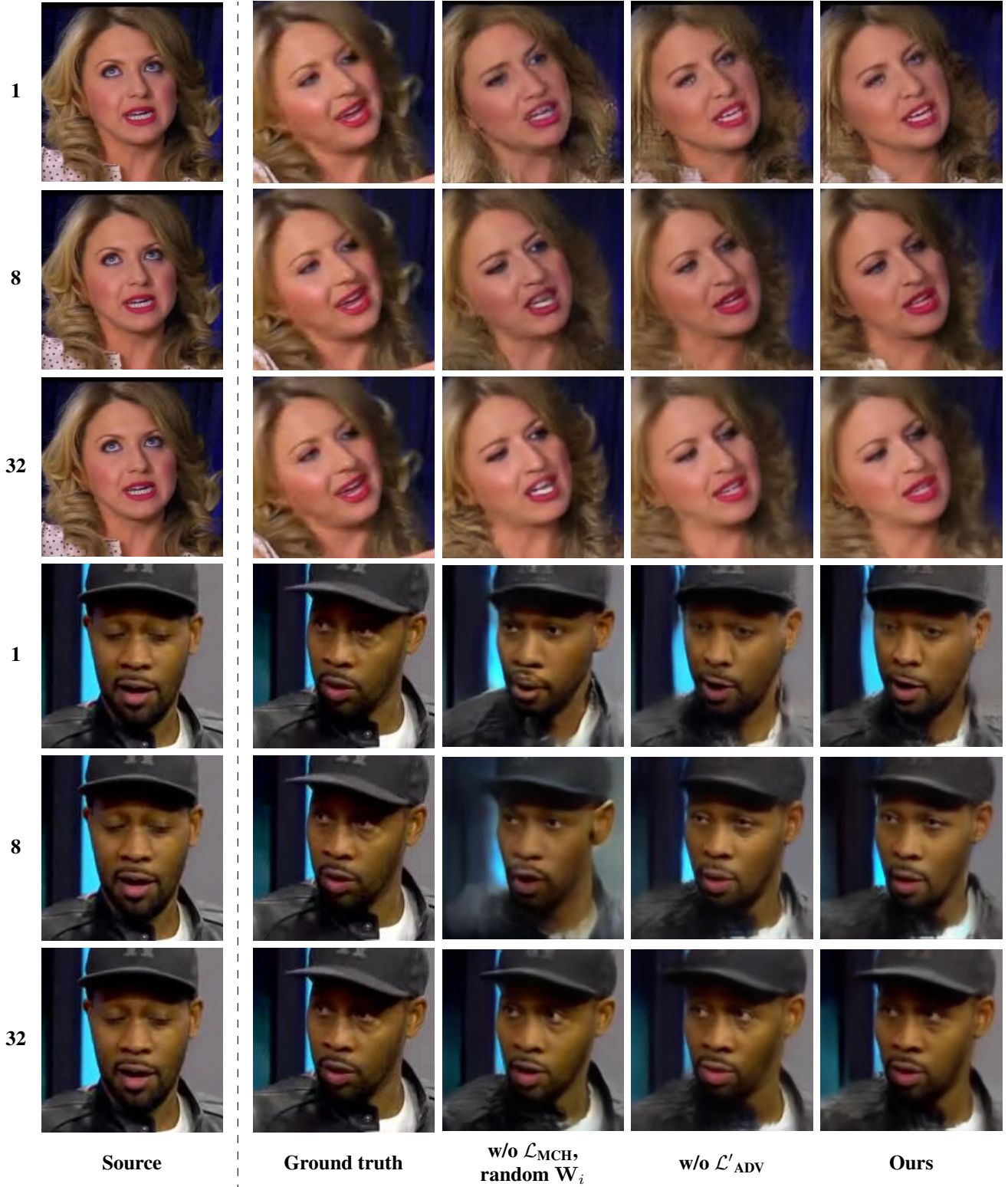
12

Source	Ground truth	w/o $\mathcal{L}_{\mathbf{MCH}}$, random $\mathbf{W}_i$	w/o $\mathcal{L}'_{\mathbf{ADV}}$	Ours

Figure 7: Ablation study of our contributions. The number of training frames is, again, equal to **T** (the leftmost column), the example training frame in shown in the **source** column and the next column shows **ground truth** image. Then, we remove $\mathcal{L}_{\mathbf{MCH}}$ from the meta-learning objective and initialize the embedding vector of the discriminator **randomly** (third column) and evaluate the contribution of adversarial fine-tuning compared to the regular fine-tuning **with no** $\mathcal{L}'_{\mathbf{ADV}}$ in the objective (fifth column). The last column represents results from our final model.
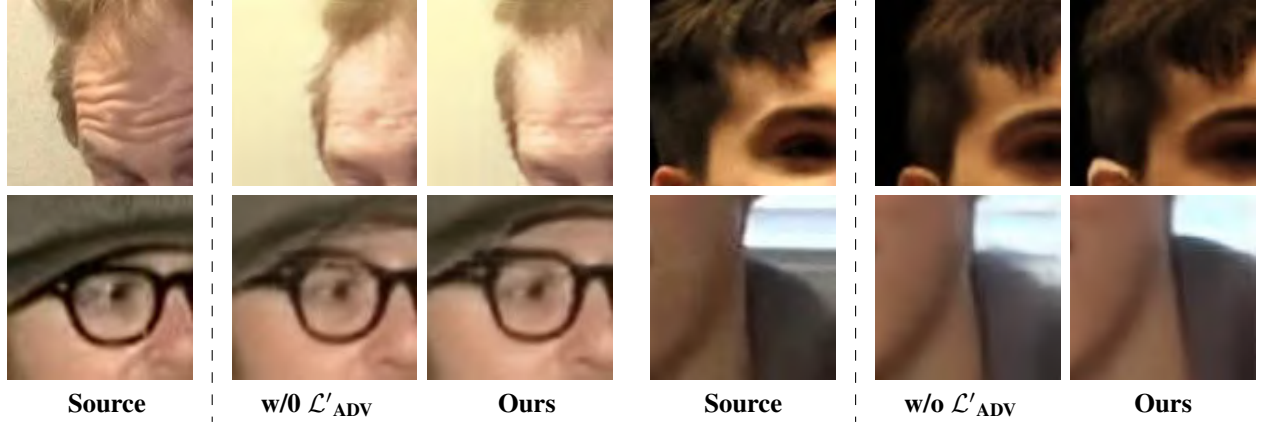
| | Source | w/0 $\mathcal{L}'_{\mathbf{ADV}}$ | Ours | Source | w/o $\mathcal{L}'_{\mathbf{ADV}}$ | Ours |

Figure 8: More close-up examples of the ablation study examples for the comparison against the model **w/o** $\mathcal{L}'_{\mathbf{ADV}}$. We used 8 training frames. Notice the geometry gap (top row) and additional artifacts (bottom row) introduced by the removal of $\mathcal{L}'_{\mathbf{ADV}}$ during fine-tuning.

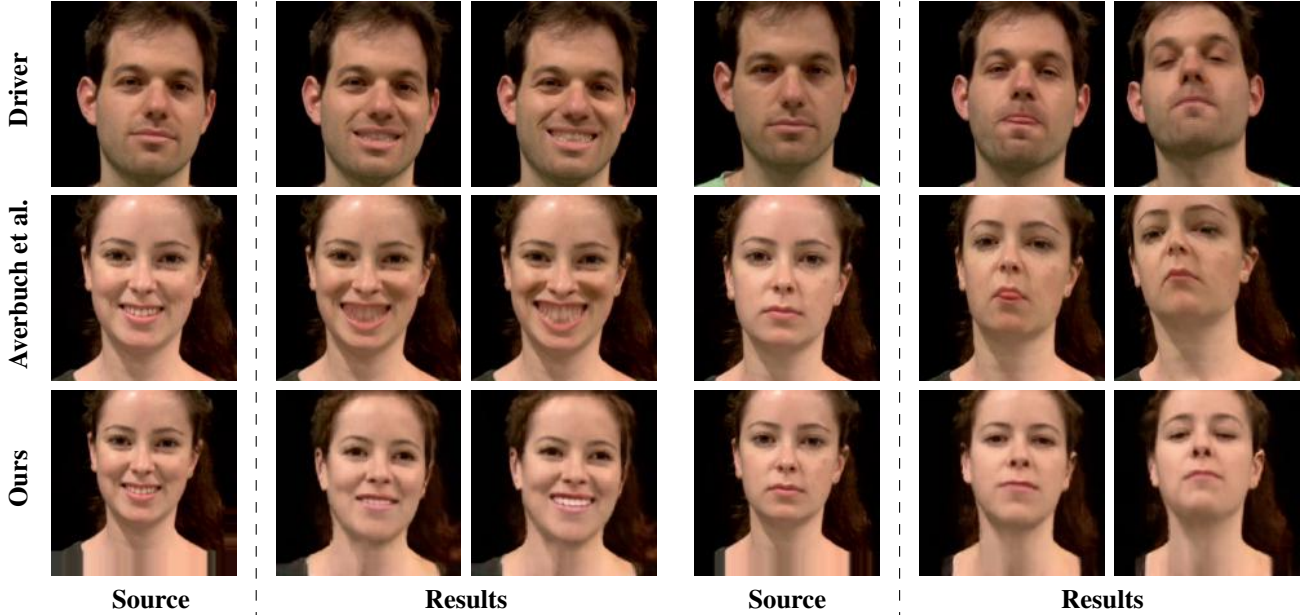| | Source | Results | Source | Results |

Figure 9: Comparison with Averbuch-Elor et al. [4] on the failure cases mentioned in the paper. Notice that our model better transfers the input pose and also is unaffected by the pose of the original frame, which lifts the "neutral face" constraint on the source image assumed in [4].

| | Source | GANimation | Our driving results |

Figure 10: Comparison with Pumarola et al. [29] (second column) and our method (right four columns). We perform the driving in the same way as we animate still images in the paper. Note that in the VoxCeleb datasets face cropping have been performed differently, so we had to manually crop our results, effectively decreasing the resolution.

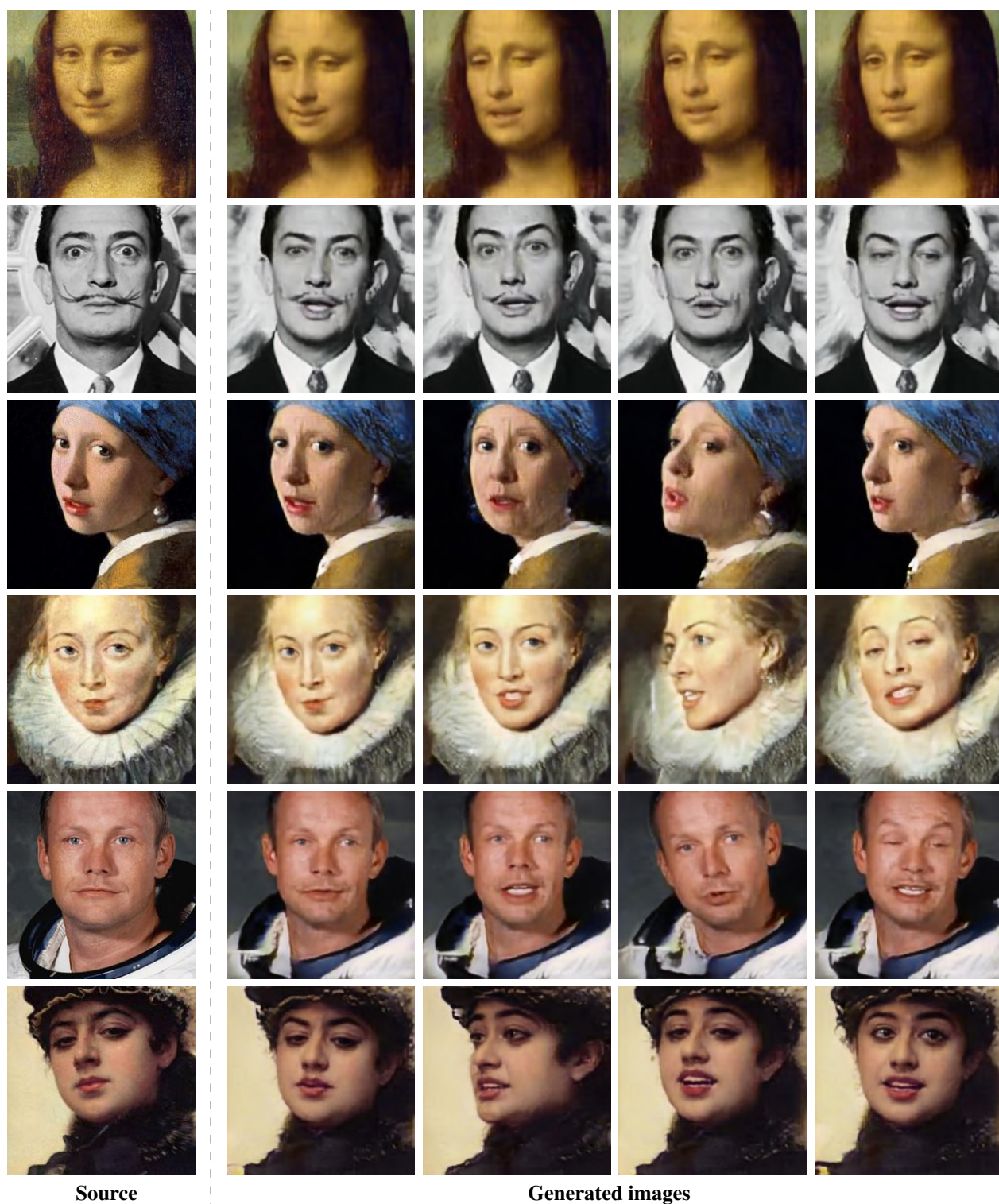**Source**   **Generated images**

Figure 11: More puppeteering results for talking head models trained in one-shot setting. The image used for one-shot training problem is in the **source** column. The next columns show **generated images**, which were conditioned on the video sequence of a different person.
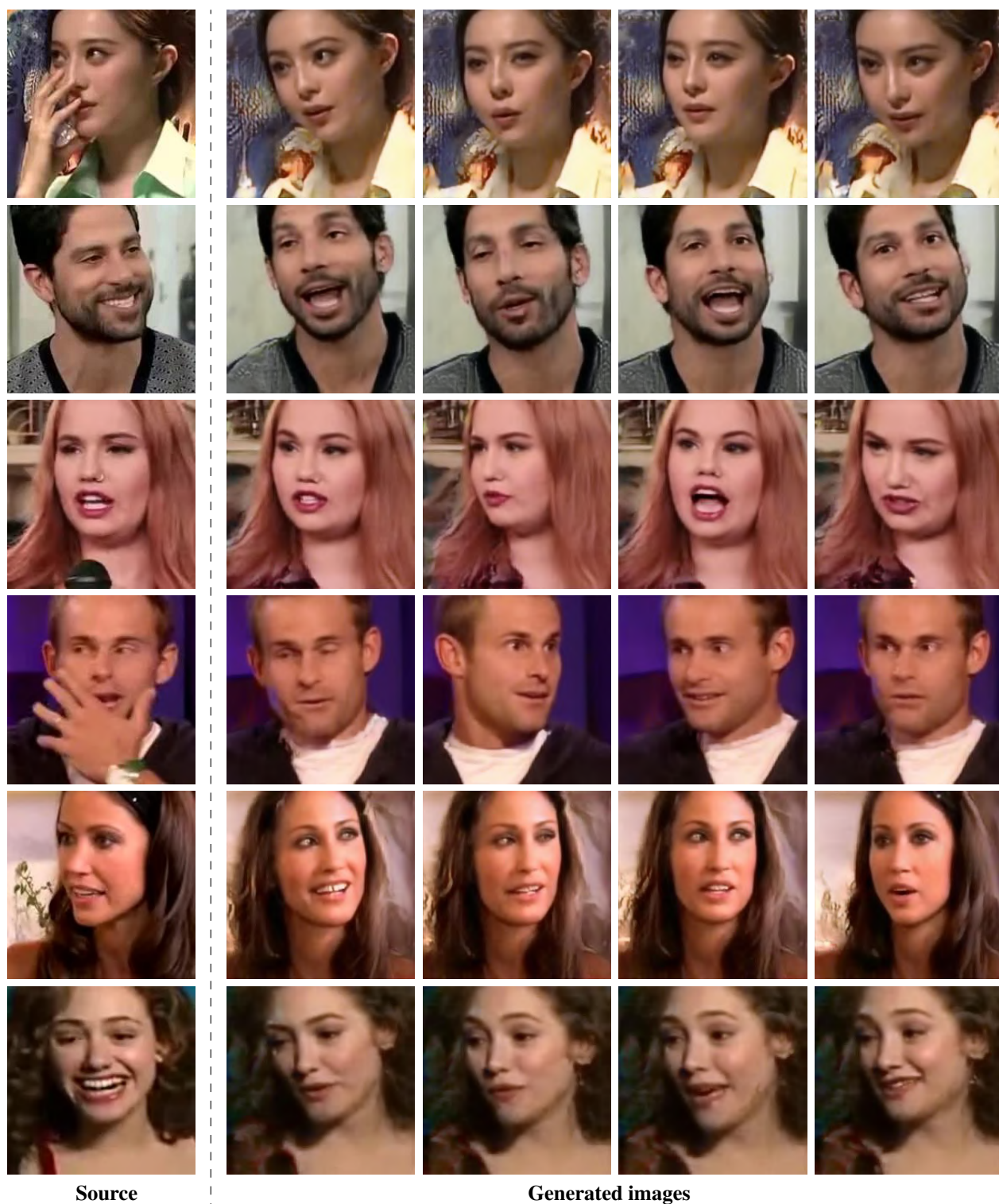
Source **Generated images**

Figure 12: Results for talking head models trained in eight-shot setting. Example training frame is in the **source** column. The next columns show **generated images**, which were conditioned on the pose tracks taken from a different video sequence with the same person.
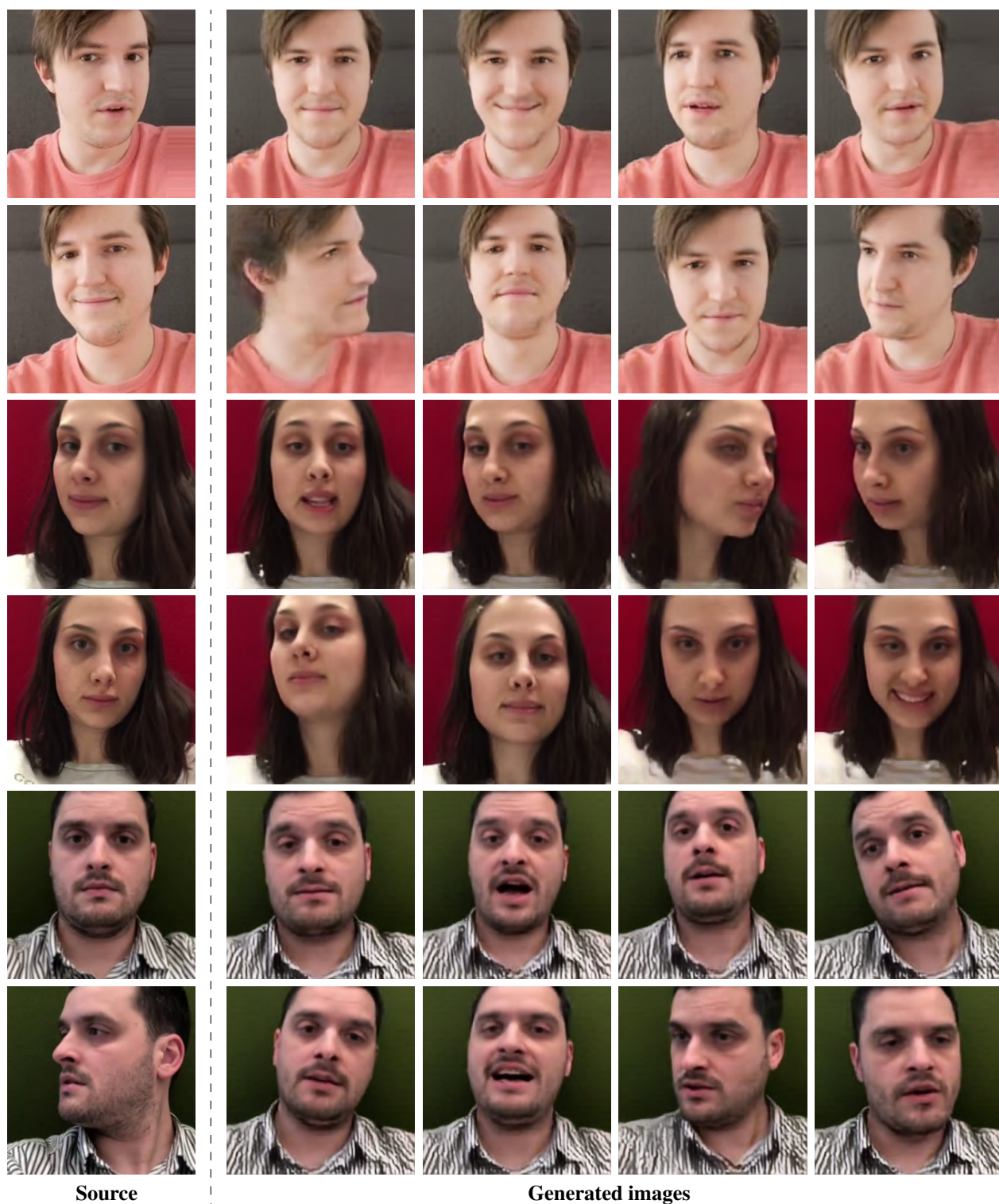
Source   **Generated images**

Figure 13: Results for talking head models trained in 16-shot setting on selfie photographs with driving landmarks taken from the different video of the same person. Example training frames are shown in the **source** column. The next columns show **generated images**, which were conditioned on the different video sequence of the same person.

T	**Source**	**Ground truth**	**X2Face**	**Pix2pixHD**	**Ours**

Figure 14: First of the extended qualitative comparisons on the VoxCeleb1 dataset. Here, the comparison is carried out with respect to both the qualitative performance of each method and the way the amount of the training data affects the results. The notation for the columns follows Figure 3 in the main paper.

Source	**Ground truth**	**X2Face**	**Pix2pixHD**	**Ours**

Figure 15: Second extended qualitative comparison on the VoxCeleb1 dataset. Here, we compare qualitative performance of the three methods on different people not seen during meta-learning or pretraining. We used eight shot learning problem formulation. The notation for the columns follows Figure 3 in the main paper.

| | Source | Ground truth | Ours-FF | Ours-FT before fine-tuning | Ours-FT after fine-tuning |

T   **Source**   **Ground truth**   **Ours-FF**   **Ours-FT before fine-tuning**   **Ours-FT after fine-tuning**
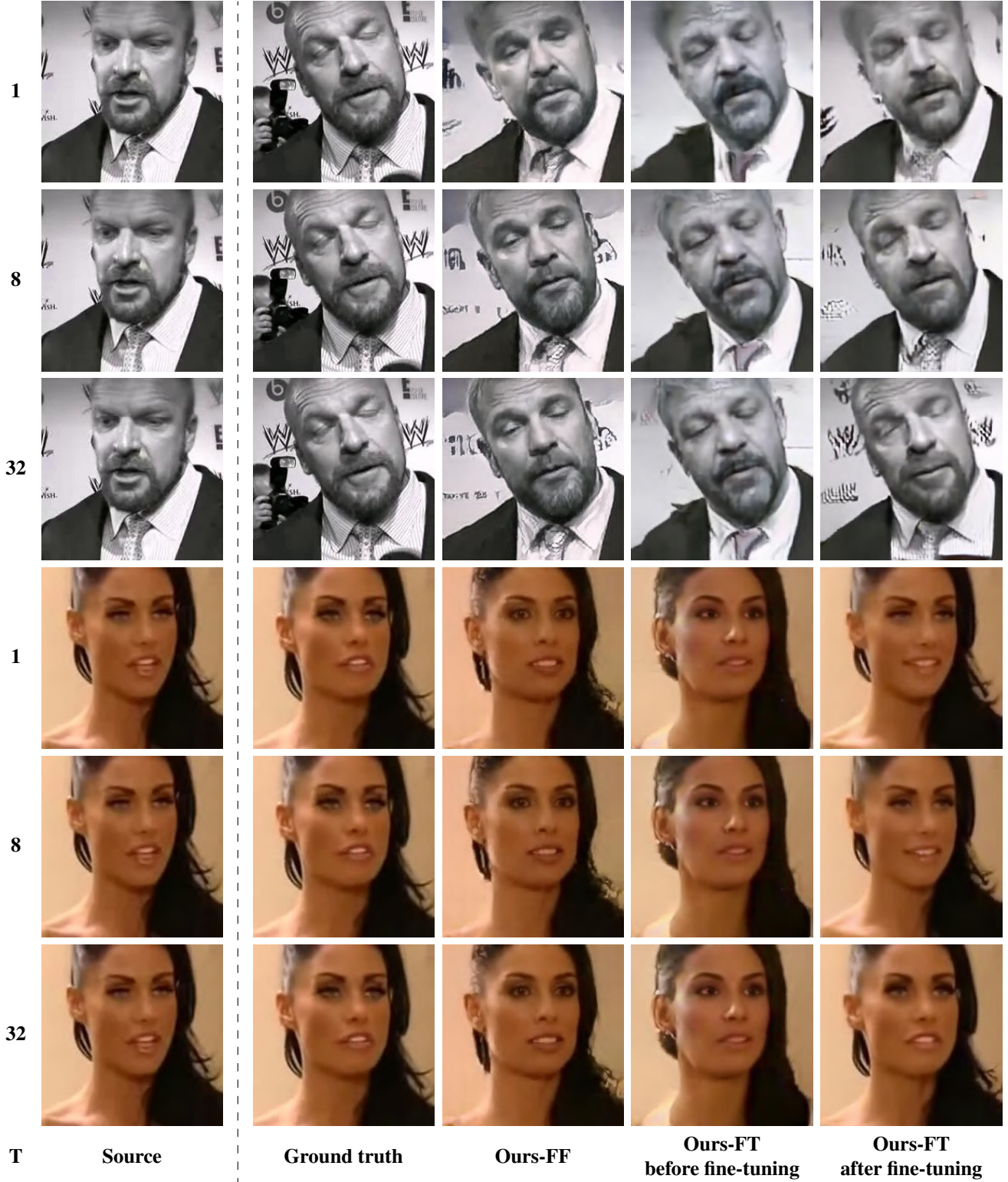
Figure 16: First of the extended qualitative comparisons on the VoxCeleb2 dataset. Here, the comparison is carried out with respect to both the qualitative performance of each variant of our method and the way the amount of the training data affects the results. The notation for the columns follows Figure 4 in the main paper.

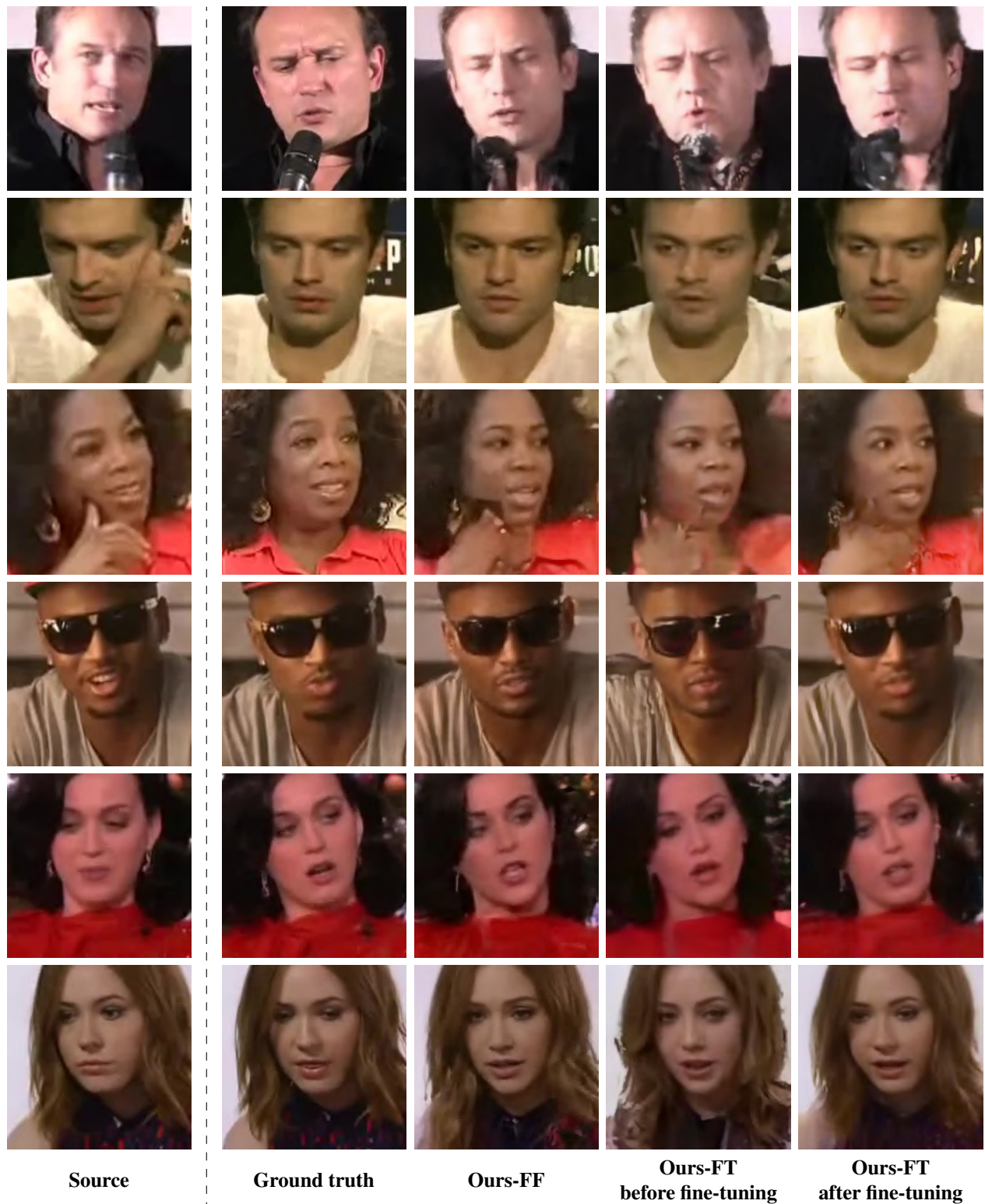| **Source** | **Ground truth** | **Ours-FF** | **Ours-FT** **before fine-tuning** | **Ours-FT** **after fine-tuning** |

Figure 17: Second extended qualitative comparison on the VoxCeleb2 dataset. Here, we compare qualitative performance of the three variants of our method on different people not seen during meta-learning or pretraining. We used eight shot learning problem formulation. The notation for the columns follows Figure 4 in the main paper.