Few-Shot Adversarial Learning of Realistic Neural Talking Head Models

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).

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.

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 $\hat{\mathbf{e}}_{\text{NEW}}$ 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.

Then, we evaluated the contribution of the person-specific initialization of the discriminator. We remove $\mathcal{L}_{\text{MCH}}$ 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 $\mathcal{L}'_{\text{ADV}}$ 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 6: Comparison with Thies et al.[36]. 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.

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.

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 [43] 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 [37], 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.

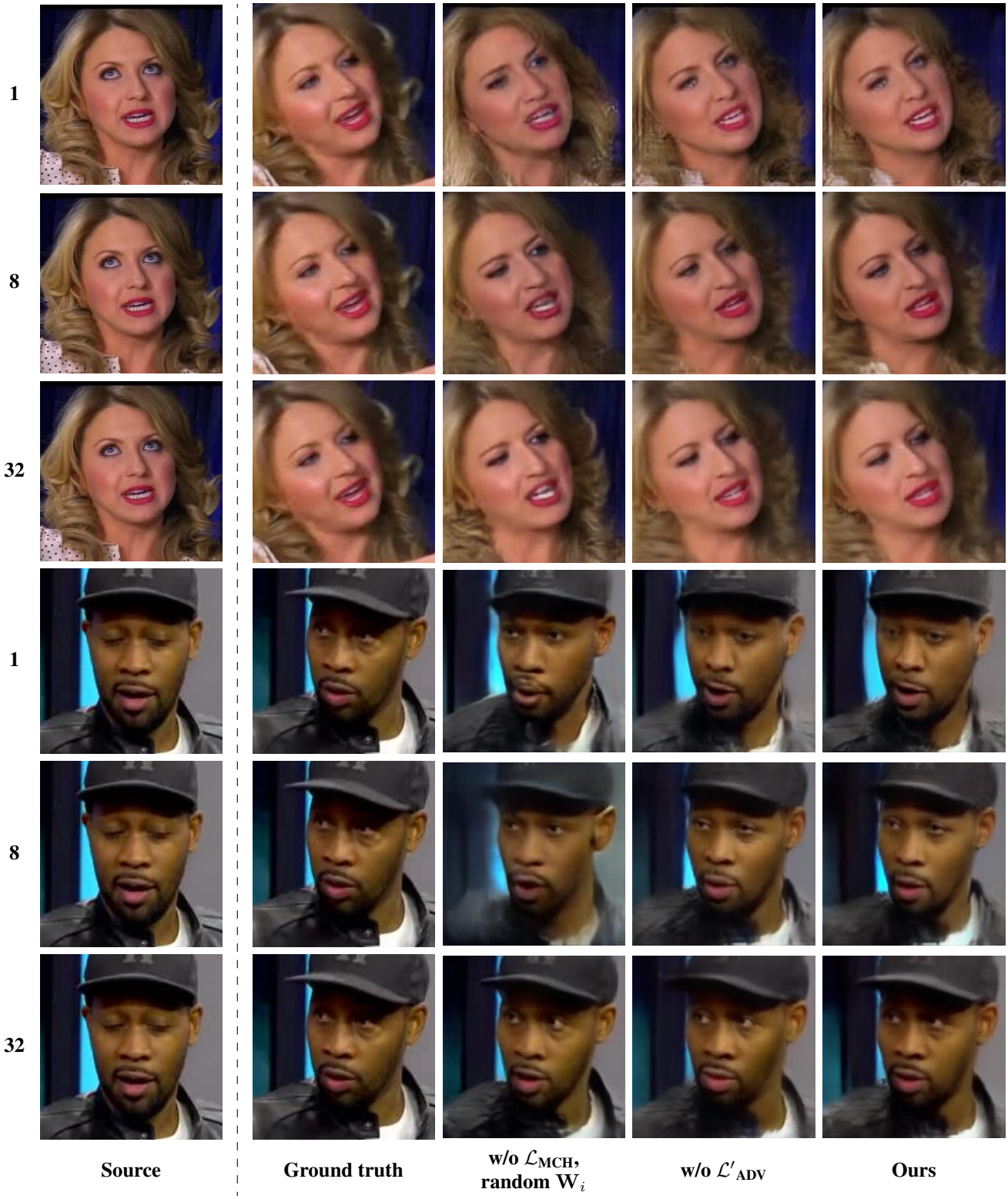

Figure 7: Ablation study of our contributions. The number of training frames is, again, equal to **T** (the leftmost column), the example training frame in shown in the **source** column and the next column shows **ground truth** image. Then, we remove $\mathcal{L}_{\text{MCH}}$ from the meta-learning 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** $\mathcal{L}'_{\text{ADV}}$ in the objective (fifth column). The last column represents results from our final model.

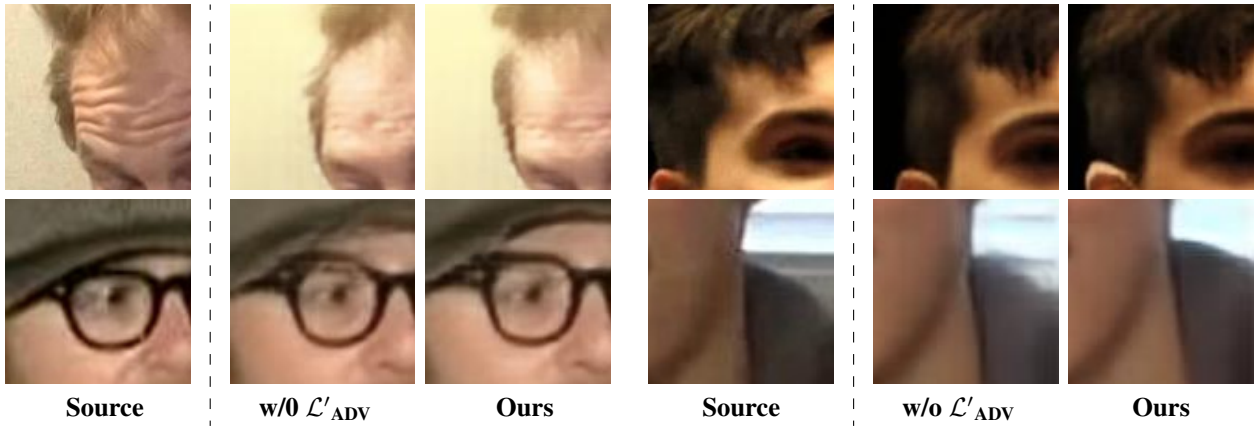

Figure 8: More close-up examples of the ablation study examples for the comparison against the model **w/o** $\mathcal{L}'_{\text{ADV}}$. We used 8 training frames. Notice the geometry gap (top row) and additional artifacts (bottom row) introduced by the removal of $\mathcal{L}'_{\text{ADV}}$ during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

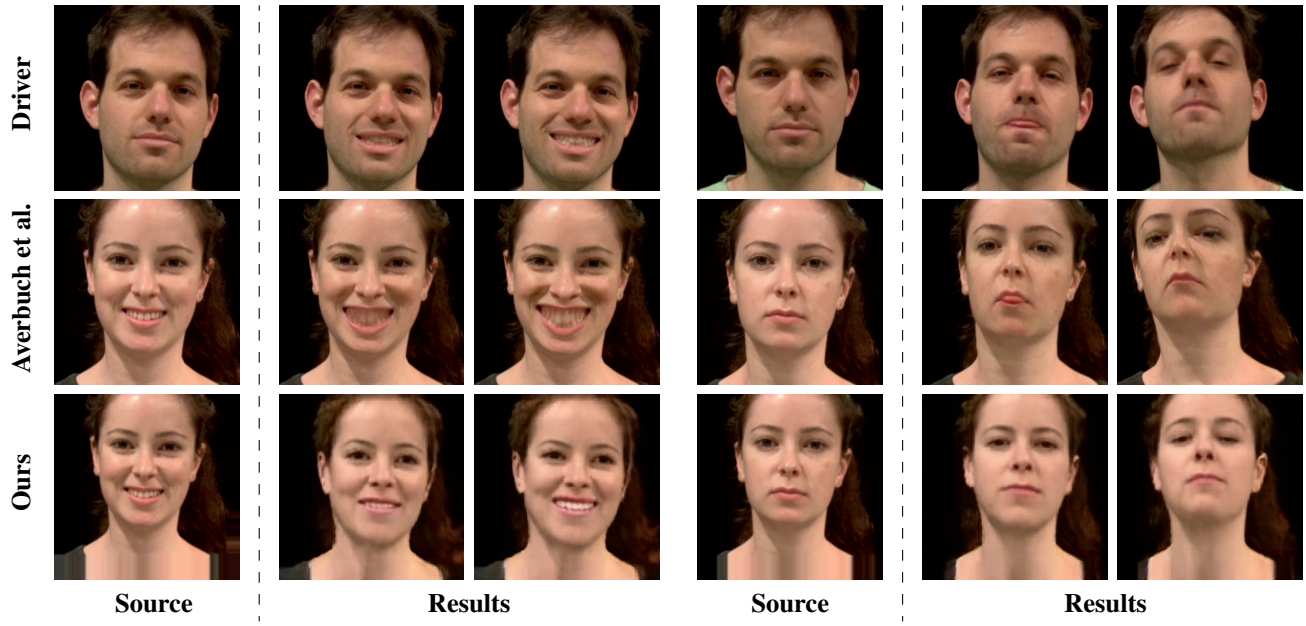

Figure 9: Comparison with Averbuch-Elor et al. [4] on the failure cases mentioned in the paper. Notice that our model better transfers the input pose and also is unaffected by the pose of the original frame, which lifts the "neutral face" constraint on the source image assumed in [4].

Figure 10: Comparison with Pumarola et al. [28] (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.

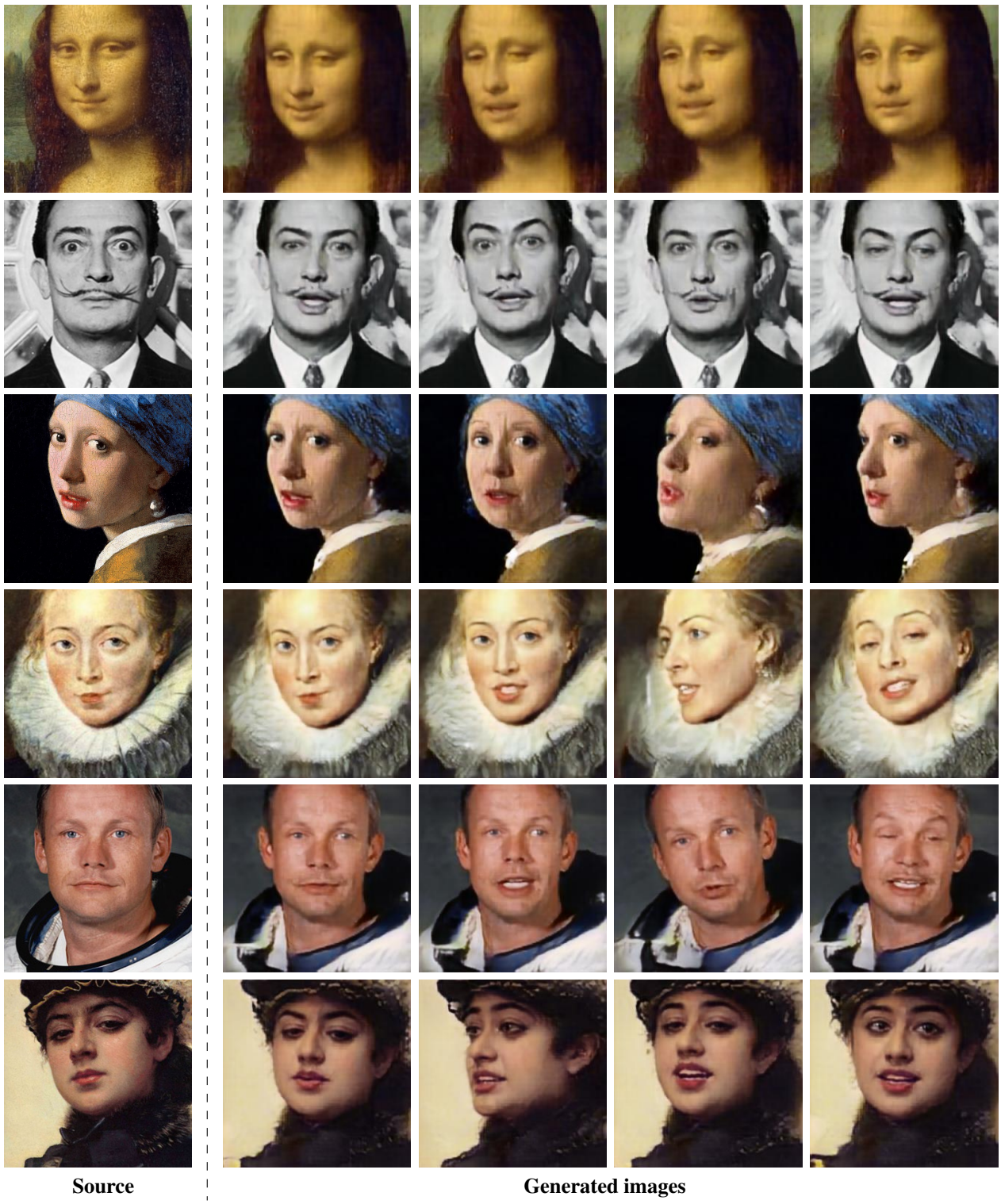

Figure 11: More puppeteering results for talking head models trained in one-shot setting. The image used for one-shot training problem is in the **source** column. The next columns show **generated images**, which were conditioned on the video sequence of a different person.

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.

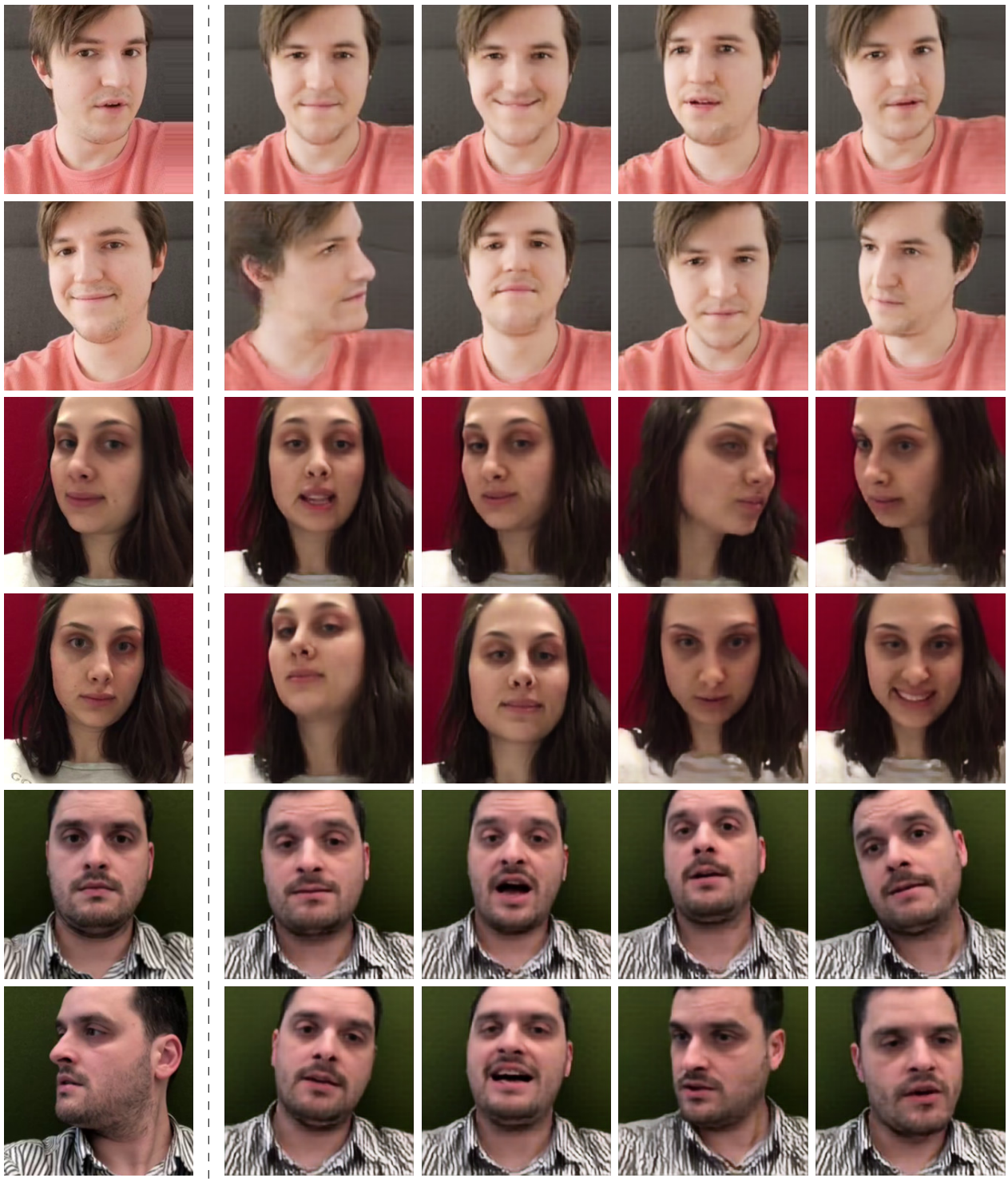

Figure 13: Results for talking head models trained in 16-shot setting on selfie photographs with driving landmarks taken from the different video of the same person. Example training frames are shown in the **source** column. The next columns show **generated images**, which were conditioned on the different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

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.

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.

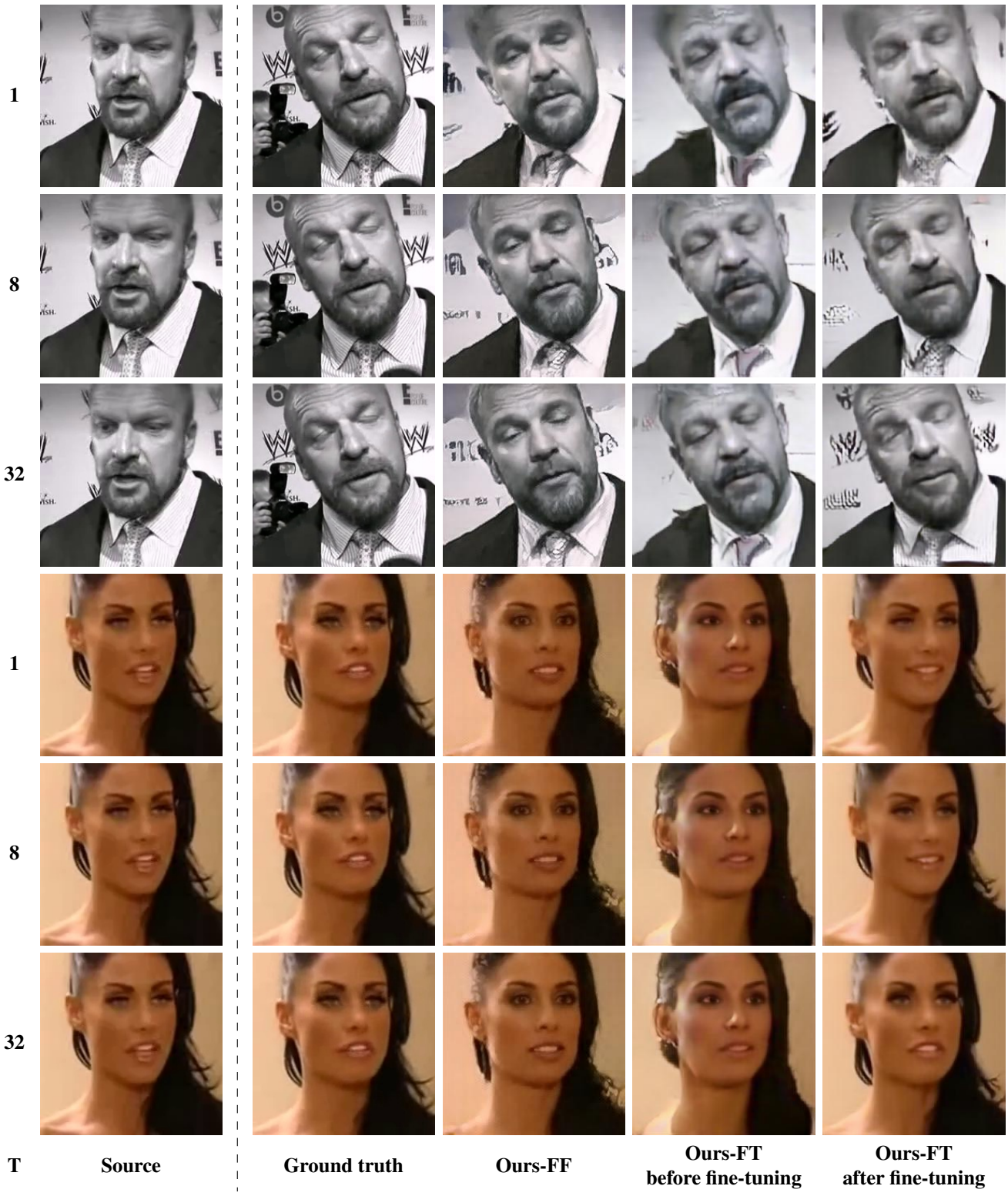

Figure 16: First of the extended qualitative comparisons on the VoxCeleb2 dataset. Here, the comparison is carried out with respect to both the qualitative performance of each variant of our method and the way the amount of the training data affects the results. The notation for the columns follows Figure 4 in the main paper.

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.