MegaPortraits: One-shot Megapixel Neural Head Avatars

Nikita Drobyshev
Samsung AI Center - Moscow
Russia

Jenya Chelishev
Samsung AI Center - Moscow
Russia

Taras Khakhulin
Samsung AI Center - Moscow
Skolkovo Institute of Science and Technology
Russia

Aleksei Ivakhnenko
Samsung AI Center - Moscow
Russia

Victor Lempitsky
Yandex
Armenia

Egor Zakharov
Samsung AI Center - Moscow
Skolkovo Institute of Science and Technology
Russia

Figure 1: We present the first system capable of creating megapixel avatars from single portrait images. Our method outperforms its competitors in the quality of the cross-driving results and manages to preserve the high-resolution appearance of the source image even for out-of-domain examples like paintings, as seen in this example.

ABSTRACT

In this work, we advance the neural head avatar technology to the megapixel resolution while focusing on the particularly challenging task of *cross-driving* synthesis, i.e., when the appearance of the *driving* image is substantially different from the animated *source* image. We propose a set of new neural architectures and training methods that can leverage both medium-resolution video data and high-resolution image data to achieve the desired levels of rendered image quality and generalization to novel views and motion. We demonstrate that suggested architectures and methods produce convincing high-resolution neural avatars, outperforming the competitors in the cross-driving scenario. Lastly, we show how a trained high-resolution neural avatar model can be distilled into a lightweight student model which runs in real-time and locks the identities of neural avatars to several dozens of pre-defined source images. Real-time operation and identity lock are essential for many practical applications head avatar systems. MegaPortraits website

CCS CONCEPTS

• **Computing methodologies** → **Image-based rendering**; *Neural networks.*

KEYWORDS

Neural rendering, generative models, one-shot neural avatars

ACM Reference Format:
Nikita Drobyshev, Jenya Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor Lempitsky, and Egor Zakharov. 2022. MegaPortraits: One-shot Megapixel Neural Head Avatars. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3503161.3547838

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10.
https://doi.org/10.1145/3503161.3547838

1 INTRODUCTION

Neural head avatars [3, 8, 12, 17–19, 22, 23, 25, 26, 30, 34, 38, 39] offer a new fascinating way of creating virtual head models. They bypass the complexity of realistic physics-based modeling of human avatars by learning the shape and appearance directly from the videos of talking people. Over the last several years, methods that can create realistic avatars from a single photograph (one-shot) have been developed [8, 26, 34, 38]. They leverage extensive pre-training on the large datasets of videos of different people [4, 34] to create the avatars in the one-shot mode using generic knowledge about human appearance.

Despite the impressive results obtained by this class of methods, their quality is severely limited by the resolution of the training datasets. This limitation cannot be easily bypassed by collecting a higher resolution dataset since it needs to be simultaneously large-scale and diverse, i.e., include thousands of humans with multiple frames per person, diverse demographics, lighting, background, face expression, and head pose. To the best of our knowledge, all public datasets [4, 34] that meet these criteria are limited in resolution. As a result, even the most recent one-shot avatar systems [34] learn the avatars at resolutions up to 512×512.

In our work, we make three main contributions. First, we propose a new model for one-shot neural avatars that achieves state-of-the-art cross-reenactment quality in up to 512×512 resolution. In our architecture, we utilize the idea of representing the appearance of the avatars as a latent 3D volume [34] and propose a new way to combine it with the latent motion representations [3], which includes a novel contrastive loss that allows our system to achieve higher degrees of disentanglement between the latent motion and appearance representations. On top of that, we add a problem-specific *gaze* loss that increases the realism and accuracy of eye animation.

Our second and crucial contribution is showing how a model trained on medium-resolution videos can be "upgraded" to the megapixel (1024×1024) resolution using an additional dataset of high-resolution still images. As a result, our proposed method, while using the same training dataset, outperforms the baseline super-resolution approach [37] for the task of cross-reenactment. We are thus the first to demonstrate neural head avatars in proper megapixel resolution.

Lastly, since many practical applications for human avatar creation require real-time or faster than real-time rendering, we distill our megapixel model into ten times faster student model that runs at 130 FPS on a modern GPU. This significant speedup is possible since the student is trained for specific appearances (unlike the main model that can create new avatars for previously unseen people). Furthermore, the applications based on such a student model "locked" to predefined identities can prevent its misuse for creating "deep fakes" while at the same time achieving low rendering latency.

2 RELATED WORK

The recent success of neural implicit scene representations [21] for the problem of 3D reconstruction has inspired several works on the so-called 4D head avatars [10, 18, 19, 22, 23, 36], which treat the problem of appearance and motion modeling of the avatars as a non-rigid reconstruction of the training video. These methods have different ways of handling the non-rigidity of motion and either learn it from scratch [22, 23, 36], use pre-trained motion extractors [10] or pre-computed coarse meshes [18, 19]. While all these methods can achieve an impressive realism of renders and fidelity of motions, they require multi-shot training data, are trained separately for each avatar, and often fail to represent motions unseen during training. In contrast, our method can impose motion from an arbitrary video sequence on an appearance obtained from a single image while still achieving megapixel resolutions of the renders.

Direct generation of videos via convolutional neural networks, conditioned on appearance and motion descriptors, is an alternative approach to talking-head synthesis. While the early works in this area learned an avatar from the video [17, 30], the follow-up works added few-shot and one-shot capabilities [3, 8, 25, 26, 34, 38, 39]. Most of these works use explicit representations for the motion, such as keypoints or blendshapes, while others [3] have adopted latent motion parameterization. The latter achieves better expressiveness of motion if the disentanglement from the appearance is achieved during training. In our system, we chose the latter approach and proposed a new method of disentangling the motion and the appearance descriptors, which significantly improves the quality of the results.

The resolution of the talking head models is currently upper bounded by the available video datasets [4, 34], which contain videos of at most 512×512 resolution. This problem further restricts the enhancement of the output quality on the existing datasets using the standard high-quality image and video synthesis techniques [32, 33]. Alternatively, this problem could be treated as single image super-resolution (SISR). This way, we require only the dataset of still high-resolution images for training, which is easier to obtain. However, the quality of the outputs of the one-shot talking head model varies greatly depending on the imposed motion, which results in poor performance of standard SISR methods [37]. These classic approaches rely on supervised training procedures with an a priori known ground truth, which we cannot provide for the novel motion data since we only have one image per person. We address this problem in a novel way by combining supervised and unsupervised training and achieve considerably better performance for arbitrary motion data than the solution based on SISR.

3 METHOD

We propose a system for the one-shot creation of high-resolution human avatars, called *megapixel portraits* or MegaPortraits for short. Our model is trained in two stages. Optionally, we propose an additional distillation stage for faster inference. Our training setup is relatively standard. We sample two random frames from our dataset at each step: the source frame $\mathbf{x}_s$ and the driver frame $\mathbf{x}_d$. Our model imposes the motion of the driving frame (i.e., the head pose and the facial expression) onto the appearance of the source frame to produce an image $\hat{\mathbf{x}}_{s \to d}$. The main learning signal is obtained from the training episodes where the source and the driver frames come from the same video, and hence our model's prediction is trained to match the driver frame. In this section, we will focus on the principal training regime while leaving details of the architectures to the supplementary materials.

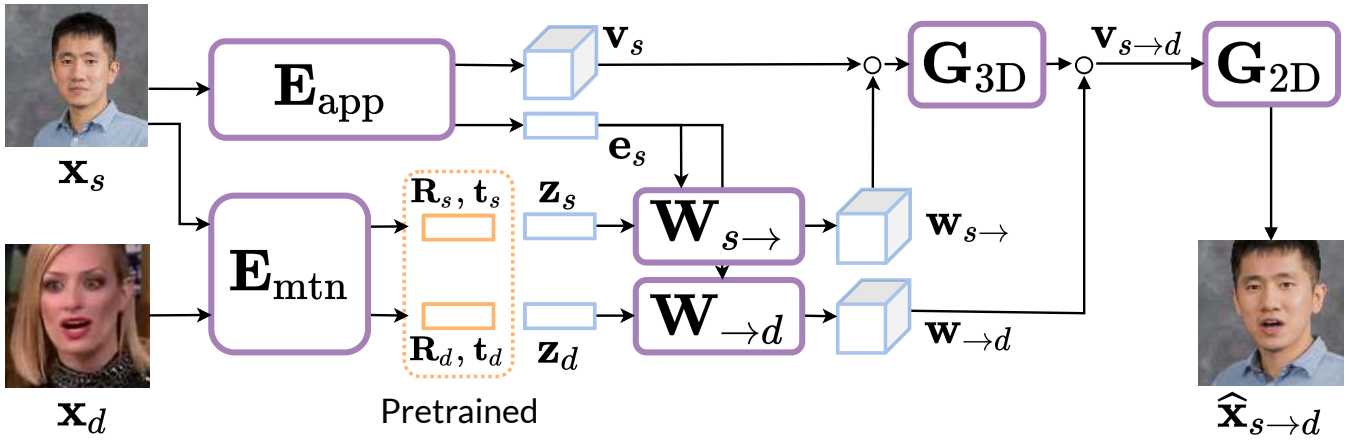


Figure 2: Overview of our base model. To encode the appearance of the source frame, we predict volumetric features $\mathbf{v}_s$ and a global descriptor $\mathbf{e}_s$ from the source image via an appearance encoder $\mathbf{E}_{\text{app}}$. In parallel, we predict the motion representations from both the source and driving images using a motion encoder $\mathbf{E}_{\text{mtn}}$. These representations consist of the explicit head rotations $\mathbf{R}_{s/d}$, translations $\mathbf{t}_{s/d}$, and the latent expression descriptors $\mathbf{z}_{s/d}$. They are used to predict the 3D warpings $\mathbf{w}_{s\to}$ and $\mathbf{w}_{\to d}$ via the separate warping generators $\mathbf{W}_{s\to}$ and $\mathbf{W}_{\to d}$. The first warping removes the source motion from the appearance features $\mathbf{v}_s$ by mapping them into a canonical coordinate space, and the second one imposes the driver motion. The canonical volume is processed by a 3D convolutional network $\mathbf{G}_{\text{3D}}$, and the driving volume $\mathbf{v}_{s\to d}$ is orthographically projected into 2D features and processed by a 2D convolutional network $\mathbf{G}_{\text{2D}}$, which predicts an output image $\hat{\mathbf{x}}_{s\to d}$.

3.1 Base model

During the first stage, we train our base model (Figure 2) by sampling two frames $\mathbf{x}_s$ and $\mathbf{x}_d$ from a random training video. The driving frame acts as both an input for our system and the ground truth. The source frame $\mathbf{x}_s$ is passed through an *appearance encoder* $\mathbf{E}_{\text{app}}$, which outputs local volumetric features $\mathbf{v}_s$ (a 4D tensor with the fourth dimension corresponding to channels), and the global descriptor $\mathbf{e}_s$. In parallel, the motion descriptors of the source and driver images are calculated by separately applying a *motion encoder* $\mathbf{E}_{\text{mtn}}$ to each image. This encoder outputs head rotations $\mathbf{R}_{s/d}$, translations $\mathbf{t}_{s/d}$, and latent expression descriptors $\mathbf{z}_{s/d}$. The source tuple $(\mathbf{R}_s, \mathbf{t}_s, \mathbf{z}_s, \mathbf{e}_s)$ is then input into a warping generator $\mathbf{W}_{s\to}$ to produce a 3D warping field $\mathbf{w}_{s\to}$, which removes the motion data from the volumetric features $\mathbf{v}_s$ by mapping them into a canonical coordinate space. These features are then processed by a 3D convolutional network $\mathbf{G}_{\text{3D}}$. Finally, the driver tuple $(\mathbf{R}_d, \mathbf{t}_d, \mathbf{z}_d, \mathbf{e}_s)$ is fed into a separate warping generator $\mathbf{W}_{\to d}$, which output $\mathbf{w}_{\to d}$ is used to impose the driver motion. The final 4D volumetric features are therefore obtained in the following way:

$$\mathbf{v}_{s\to d} = \mathbf{w}_{\to d} \circ \mathbf{G}_{\text{3D}}(\mathbf{w}_{s\to} \circ \mathbf{v}_s), \tag{1}$$

where $\circ$ represents a 3D warping operation. The idea behind this approach is first to rotate the volumetric features into a frontal viewpoint, remove any face expression motion decoded from $\mathbf{z}_s$, process them by a 3D convolutional network, and then impose the driver head rotation and motion. We use a pre-trained network to estimate head rotation data, but the latent expression vectors $\mathbf{z}_{s/d}$ and the warpings to and from the canonical coordinate space are trained without direct supervision.

The volumetric feature encoding and the explicit use of head pose are inspired by [34]. However, a significant difference with [34] is that we do not use keypoints to represent expression and instead rely on the latent descriptor [3], which is decoded into the explicit 3D warping field to represent face mimics in a more person-independent way. We have also observed that the motion disentanglement scheme proposed in [3] starts to fail when we increase the capacity of the avatar system to facilitate higher resolutions. This problem leads to severe appearance leakage from the driving to the predicted image. To combat that, we propose using a cycle-consistency loss, which we describe below, and improving the driving image's pre-processing pipeline. For more details, please refer to the supplementary materials.

Finally, the driver volumetric features $\mathbf{v}_{s\to d}$ are orthographically projected into the camera frame using the same approach as in [34]. We denote this operation as $\mathcal{P}$. The resulting 2D feature map is decoded into the output image by a 2D convolutional network $\mathbf{G}_{\text{2D}}$:

$$\hat{\mathbf{x}}_{s\to d} = \mathbf{G}_{\text{2D}}(\mathcal{P}(\mathbf{v}_{s\to d})). \tag{2}$$

We refer to the combination of the networks described above as $\mathbf{G}_{\text{base}}$, so that

$$\hat{\mathbf{x}}_{s\to d} = \mathbf{G}_{\text{base}}(\mathbf{x}_s, \mathbf{x}_d). \tag{3}$$

We use multiple loss functions for training, which can be split into two groups. The first group consists of the standard training objectives for image synthesis. These include perceptual [14] and GAN [33] losses that match the predicted image $\hat{\mathbf{x}}_{s\to d}$ to the ground-truth $\mathbf{x}_d$. The other objective regularizes the training and introduces disentanglement between the motion and canonical space appearance features via the cycle consistency [42] loss.

Perceptual losses match the motion and appearance of the predicted image $\hat{\mathbf{x}}_{s\to d}$ to the ground-truth $\mathbf{x}_d$. We use three types of pre-trained networks for the perceptual losses: regular ILSVRC (ImageNet) [6] pre-trained VGG19 [27] to match the general content of

the images, VGGFace [24] trained for face recognition to match the facial appearance, and a specialized gaze loss based on VGG16 to match the gaze direction. The latter network was trained to distill a state-of-the-art gaze detection system [9]. For more details on the training and usage of gaze loss, please refer to the supplementary materials. We calculate the weighted L1 distance between the feature maps obtained for the predicted $\hat{\mathbf{x}}_{s\to d}$ and ground-truth $\mathbf{x}_d$ images using all these networks. The final perceptual loss is a weighted combination of individual perceptual losses:

$$\mathcal{L}_{\text{per}} = w_{\text{IN}}\mathcal{L}_{\text{IN}} + w_{\text{face}}\mathcal{L}_{\text{face}} + w_{\text{gaze}}\mathcal{L}_{\text{gaze}}. \tag{4}$$

Adversarial losses ensure the realism of the predicted images. We calculate these losses using the same predicted and driving images. Following the previous works, we train a multi-scale patch discriminator [42] with a hinge adversarial loss alongside the generator $\mathbf{G}_{\text{base}}$. We also include a standard feature-matching loss [33] to improve the training stability. The GAN loss for the generator can therefore be expressed as follows:

$$\mathcal{L}_{\text{GAN}} = w_{\text{adv}}\mathcal{L}_{\text{adv}} + w_{\text{FM}}\mathcal{L}_{\text{FM}}. \tag{5}$$

Cycle consistency loss is used to prevent the appearance leakage through the motion descriptor. During training, this task is essential since the motion descriptor is calculated using the same image as the ground truth. Without this regularizer, severe artifacts are present when the driver differs from the source in lighting, hair and beard style, or sunglasses because these features are leaked from the driver image onto the predicted image.

In order to calculate this loss, we use an additional source-driving pair $\mathbf{x}_{s^*}$ and $\mathbf{x}_{d^*}$, which is sampled from a different video and therefore has different appearance from the current $\mathbf{x}_s$, $\mathbf{x}_d$ pair. We then apply the full base model to produce the following *cross-reenacted* image: $\hat{\mathbf{x}}_{s^*\to d} = \mathbf{G}_{\text{base}}(\mathbf{x}_{s^*}, \mathbf{x}_d)$, and also separately calculate a motion descriptor $\mathbf{z}_{d^*} = \mathbf{E}_{\text{mtn}}(\mathbf{x}_{d^*})$. Note that we will also use the stored motion descriptors $\mathbf{z}_{s^*\to d}$ and $\mathbf{z}_{s\to d}$ from the respective forward passes of the base network.

We then arrange the motion descriptors into *positive pairs* $\mathcal{P}$ that should align with each other: $\mathcal{P} = \left\{(\mathbf{z}_{s\to d}, \mathbf{z}_d), (\mathbf{z}_{s^*\to d}, \mathbf{z}_d)\right\}$, and the *negative pairs*: $\mathcal{N} = \left\{(\mathbf{z}_{s\to d}, \mathbf{z}_{d^*}), (\mathbf{z}_{s^*\to d}, \mathbf{z}_{d^*})\right\}$. These pairs are used to calculate the following cosine distance:

$$d(\mathbf{z}_i, \mathbf{z}_j) = s \cdot \left(\langle \mathbf{z}_i, \mathbf{z}_j \rangle - m\right), \tag{6}$$

where both s and m are hyperparameters. This distance is then used to calculate a large margin cosine loss (CosFace) [31]:

$$\mathcal{L}_{\text{cos}} = -\sum_{(\mathbf{z}_k, \mathbf{z}_l)\in\mathcal{P}} \log \frac{\exp\left\{d(\mathbf{z}_k, \mathbf{z}_l)\right\}}{\exp\left\{d(\mathbf{z}_k, \mathbf{z}_l)\right\} + \sum\limits_{(\mathbf{z}_i, \mathbf{z}_j)\in\mathcal{N}} \exp\left\{d(\mathbf{z}_i, \mathbf{z}_j)\right\}}. \tag{7}$$

To conclude, the total loss which is used to train the base model is the sum of individual losses:

$$\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{per}} + \mathcal{L}_{\text{GAN}} + w_{\text{cos}}\mathcal{L}_{\text{cos}}. \tag{8}$$

These losses are calculated using only foreground regions in both predictions and the ground truth. Hence, our model has no background generation built into it, which we found empirically to hinder its performance. Instead, we impose the background post-training via pre-trained inpainting and matting models. We obtain the background plate using a state-of-the-art inpainting system [29] and use the following systems for matting [11, 16]. The background is combined with the predicted image via alpha-compositing using a calculated matte. For more details, please refer to the supplementary materials.

3.2 High-resolution model

For the second training stage, we fix the base neural head avatar model $\mathbf{G}_{\text{base}}$, and only train an image-to-image translation network $\mathbf{G}_{\text{enh}}$ that maps the input $\hat{\mathbf{x}}$ at the resolution 512×512 to an *enhanced* version $\hat{\mathbf{x}}^{\text{HR}}$ that has the resolution 1024×1024. We use a high-resolution dataset of photographs [15] to train this model, in which we assume all images to have different identities. It implies that we cannot form source-driver pairs that only differ in their motion as we do in the first training stage.

The high-resolution model is trained using two groups of loss functions. The first group represents the standard super-resolution objectives, for which use an L_1 loss, denoted as $\mathcal{L}_{\text{MAE}}$, and a GAN loss $\mathcal{L}_{\text{GAN}}$. The second group of objectives works in an unsupervised way, and we use it to ensure that our model performs well for the images generated in a cross-driving scenario. To do that, for each training image $\mathbf{x}^{\text{HR}}$ we sample an additional image $\mathbf{x}_{\text{c}}^{\text{HR}}$, and generate its initial reconstruction $\hat{\mathbf{x}}_{\text{c}} = \mathbf{G}_{\text{base}}(\mathbf{x}^{\text{LR}}, \mathbf{x}_{\text{c}}^{\text{LR}})$, where $\mathbf{x}_{\text{c}}^{\text{LR}}$ is used to estimate motion, and $\mathbf{x}^{\text{LR}}$ is used to estimate appearance. Since we do not have high-resolution ground-truth for $\hat{\mathbf{x}}_{\text{c}}^{\text{HR}} = \mathbf{G}_{\text{enh}}(\hat{\mathbf{x}}_{\text{c}})$, we can only match its distribution to ground truth using a patch discriminator. Furthermore, we can enforce content preservation by applying the cycle-consistency loss at lower resolution:

$$\mathcal{L}_{\text{cyc}}^{\text{c}} = \mathcal{L}_{\text{MAE}}\left(\text{DS}_4(\hat{\mathbf{x}}_{\text{c}}), \text{DS}_8(\hat{\mathbf{x}}_{\text{c}}^{\text{HR}})\right), \tag{9}$$

where DS_k denotes a k-times downsampling operator.

The final objective for $\mathbf{G}_{\text{enh}}$ includes the adversarial and the perceptual losses calculated for the predicted image $\hat{\mathbf{x}}^{\text{HR}}$ and its ground-truth $\mathbf{x}^{\text{HR}}$, as well as an adversarial loss $\mathcal{L}_{\text{adv}}^{\text{c}}$, calculated for $\hat{\mathbf{x}}_{\text{c}}^{\text{HR}}$ and $\mathbf{x}^{\text{HR}}$, and the cycle-consistency loss $\mathcal{L}_{\text{cyc}}^{\text{c}}$:

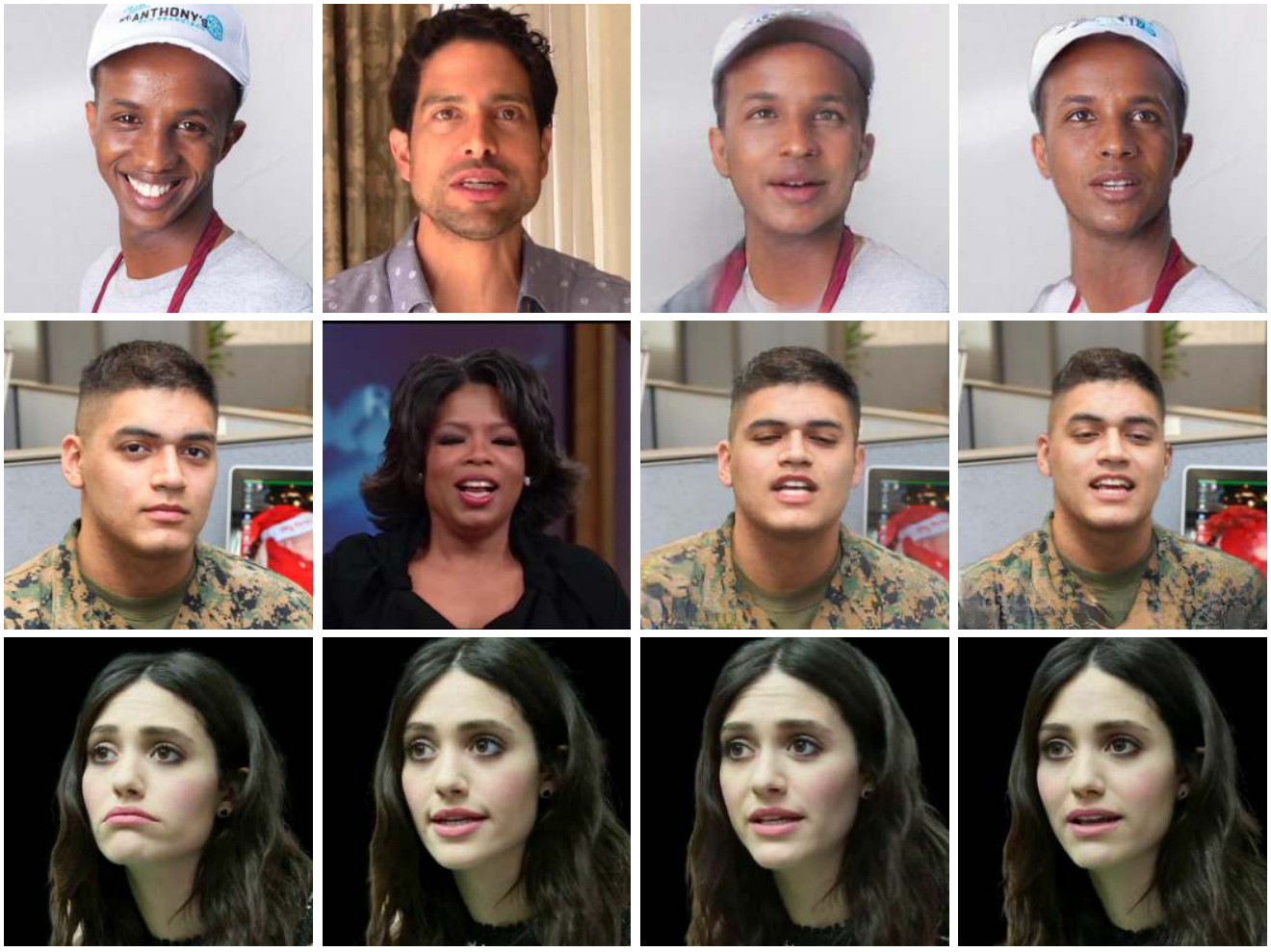
$$\mathcal{L}_{\text{enh}} = \mathcal{L}_{\text{GAN}} + w_{\text{MAE}}\mathcal{L}_{\text{MAE}} + w_{\text{adv}}^{\text{c}}\mathcal{L}_{\text{adv}}^{\text{c}} + w_{\text{cyc}}^{\text{c}}\mathcal{L}_{\text{cyc}}^{\text{c}}. \tag{10}$$

3.3 Student model

Finally, we use a small conditional image-to-image translation network $\mathbf{G}_{\text{DT}}$, which we refer to as the *student*, to distill the one-shot model. We train the student to mimic the prediction of the full (teacher) model $\mathbf{G}_{\text{HR}} = \mathbf{G}_{\text{enh}} * \mathbf{G}_{\text{base}}$, which combines the base model and an enhancer. The student is trained only in the cross-driving mode by generating pseudo-ground truth with the teacher model. Since we train our student network for a limited number of avatars, we condition it using an index i, which selects an image from the set of all N appearances $\{\mathbf{x}_i\}_{i=1}^{N}$. Therefore, training proceeds as follows: we sample the driving frame $\mathbf{x}_d$ and the index i. We then match the following two images:

$$\hat{\mathbf{x}}_{i\to d}^{\text{DT}} = \mathbf{G}_{\text{DT}}(\mathbf{x}_d, i); \qquad \hat{\mathbf{x}}_{i\to d}^{\text{HR}} = \mathbf{G}_{\text{HR}}(\mathbf{x}_i, \mathbf{x}_d).$$

We train this network using a combination of perceptual and adversarial losses. For architectural details, please refer to the supplementary materials.



Source **Driver** **Face-V2V** [34] **Ours**

Figure 3: A qualitative comparison of head avatar systems in cross-reenactment scenario (top two rows) and self-reenactment scenario (bottom row) at 512px resolution. In cross-reenactment, we can see that our approach achieves better preservation of motion and appearance than the previous state-of-the-art (Face-V2V). In self-reenactment, we achieve the results of comparable quality with the state-of-the-art. For more examples, please refer to the supplementary materials.

4 EXPERIMENTS

We use multiple datasets to train and evaluate our model: VoxCeleb2 [4] and VoxCeleb2HQ video datasets, and FFHQ [15] image dataset. We have obtained a high-quality version of the VoxCeleb2 dataset, which we refer to as VoxCeleb2HQ, by downloading the original videos and filtering them using both bitrate and image quality assessment [28]. This leaves approximately one-tenth of the original dataset (15,000 videos). We use this dataset to train and evaluate our base model at 512×512 resolution while using the original VoxCeleb2 dataset, filtered using bitrate, for the 256×256 resolution. For training a high-resolution model, we used a filtered version of the FFHQ dataset, which consists of 20,000 images and has no frames that contain multiple people or children. Lastly, we use a proprietary dataset of 20,000 selfie videos and 100,000 selfie pictures to train the student model.

4.1 Training details

We trained the 256×256 model for 200,000 iterations with the batch size of 24, and the 512×512 model for 300,000 iterations with the batch size of 16. We used AdamW [20] optimizer with cosine learning rate scheduling. The initial learning rate was reduced from $2 * 10^{-4}$ to 10^{-6} during training iterations. We used the following hyperparameters for the losses: $w_{\text{IN}} = 20$, $w_{\text{face}} = 4$, $w_{\text{gaze}} = 5$, $w_{\text{adv}} = 1$, $w_{\text{FM}} = 40$, and $w_{\text{cos}} = 2$. We also set $s = 5$ and $m = 0.2$ in the cosine loss.

We trained the high-resolution enhancer model for 50,000 iterations with the batch size of 16. We used the same optimizer and the learning rate scheduling. We set the loss weights to $w_{\text{MAE}} = 100$, $w^{\text{c}}_{\text{adv}} = 1$, $w_{\text{FM}} = 100$ and $w^{\text{c}}_{\text{cyc}} = 10$. Finally, for the student model we distilled 100 avatars. We trained it for 170,000 iterations with the batch size of 8. For detailed descriptions of all architectures, please refer to the supplementary material.

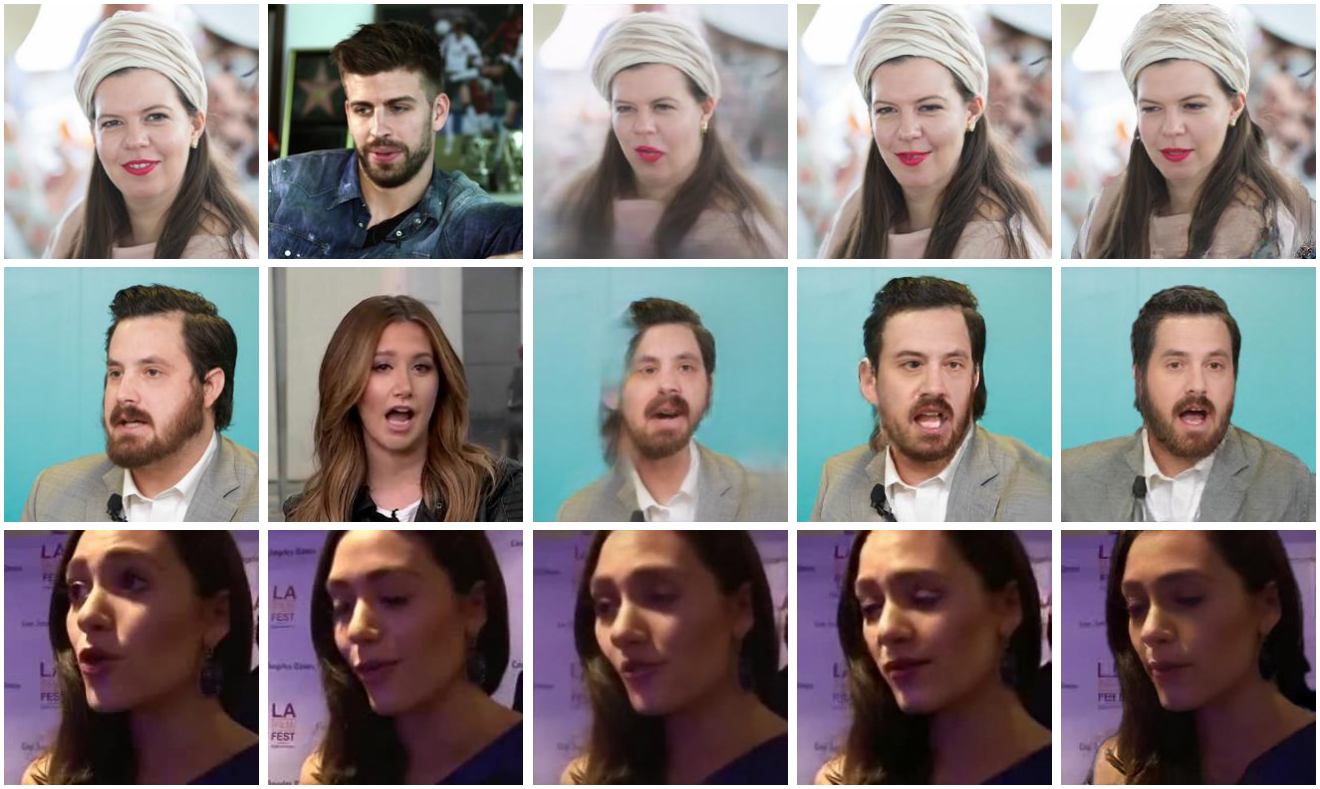


Figure 4: A qualitative comparison of head avatar systems in cross-reenactment scenario (top two rows) and self-reenactment scenario (bottom row) at 256×256 resolution. Our system significantly outperforms the competitors in cross-reenactment, achieving more faithful motion and appearance preservation in the generated images. We also show that our system achieves similar results in self-reenactment. For more examples, please refer to the supplementary materials.

4.2 Baseline methods

We compare our base model with the following systems.

Face Vid-to-vid (Face-V2V) [34] is a state-of-the-art system in self-reenactment, i.e. when the source and driving images have the same appearance and identity. Its main features are the volumetric encoding of the avatar's appearance and the explicit representation of the head motion with 3D keypoints, which are learned in an unsupervised way. In our base model, we utilize a similar volumetric encoding of the appearance but instead encode the face motion implicitly, which improves cross-reenactment performance.

First Order Motion Model (FOMM) [26] uses 2D keypoints to represent motion and is another strong baseline for the task of self-reenactment. Similar to Face-V2V, these keypoints are trained in an unsupervised way. However, as shown in our evaluation, this method fails to generate realistic images in the cross-reenactment scenario.

Lastly, we compare against the HeadGAN [8] system, in which the expression coefficients of the 3D morphable model [1] are used as a motion representation. These coefficients are calculated using a pre-trained dense 3D keypoints regressor [7]. Effectively, this approach disentangles motion data from the appearance in the 3D keypoints, but limits the space of possible motions (for example, it does not allow the control of the gaze direction).

4.3 Cross-reenactment evaluation

Since pre-trained models of FOMM and HeadGAN are only available at 256×256 resolution, we compare them against our base model trained on a bitrate-filtered VoxCeleb2 dataset. For Face-V2V, we compare the 512×512 model pre-trained on the TalkingHead-1KH [34] dataset to our base model trained on the VoxCeleb2HQ. For the evaluation, we use samples from the VoxCeleb2HQ and FFHQ datasets, downscaled to the training resolution.

For quantitative evaluation, we use the following metrics. *Frechet Inception Distance* (FID) [13] is used to compare the distributions of predicted images and the images in the dataset. *Cosine similarity* between the embeddings of a face recognition network (CSIM) [39] is used to evaluate the preservation of a person's appearance in the predicted image. Finally, we conduct two *user studies* (denoted as UMTN and UAPP) to evaluate the motion and appearance preservation. We show the crowd-sourced users a random triplet of images: a driving example to evaluate motion preservation or a source example to evaluate the appearance, alongside the outputs of two random methods. We then ask each user to pick one of the two outputs with the better-preserved motion or appearance. We then measure the percentage of examples where each method was picked. We conducted our experiment on approximately 2,000 crowd-sourced

Cross-reenactment

Method	FID↓	CSIM↑	UMTN↑	UAPP↑
VoxCeleb2HQ & FFHQ (256 × 256)				
FOMM	79.1	0.63	24.0	27.9
HeadGAN	70.0	0.66	23.6	32.1
Ours	68.9	0.72	52.4	40.0
VoxCeleb2HQ & FFHQ (512 × 512)				
Face-V2V	63.4	0.70	34.4	45.4
Ours	58.8	0.73	65.6	54.6

Self-reenactment (raw / masked)

Method	PSNR↑	SSIM↑	LPIPS↓
VoxCeleb2 (256 × 256)			
FOMM	20.6 / 27.5	0.74 / 0.90	0.18 / 0.06
HeadGAN	18.6 / 26.5	0.68 / 0.88	0.20 / 0.07
Ours	18.3 / 27.0	0.67 / 0.89	0.23 / 0.07
VoxCeleb2HQ (512 × 512)			
Face-V2V	21.9 / 31.2	0.76 / 0.90	0.18 / 0.06
Ours	20.2 / 30.2	0.72 / 0.89	0.22 / 0.07

Table 1: Quantitative results for cross and self-reenactment. To evaluate cross-reenactment performance, we measure FID (lower the better), CSIM (higher the better), and user preference scores (UMTN measures motion preservation and UAPP – appearance, both are higher the better). Our method outperforms its competitors across all metrics at both resolutions, achieving state-of-the-art results in the cross-reenactment scenario. The gap is especially noticeable in the user study, where we achieve significantly better motion preservation. We use standard PSNR, SSIM (higher the better), and LPIPS (lower the better) metrics to evaluate the self-reenactment. We measure each metric using either raw or masked images. Our method performs similarly to the competitors when face masking is applied while achieving reasonable results in the unmasked (raw) scenario.

people, and each evaluation sample was shown, on average, to twenty different users.

The qualitative results are shown in Figures 3-4, and the quantitative metrics are presented in Table 1. Overall, we can see that our method outperforms all competitors by some margin. Furthermore, the first two rows in Figure 4 suggest that our approach is better at preserving the shape and appearance of the source image and the motion of the driver image, including gaze direction, than the FOMM and HeadGAN systems. Compared to the Face V2V system (Figure 3, first two rows), our implicit pose representation approach prevents appearance leakage through the driving image, leading to better preservation of the source image appearance, as well as driver motion. These observations are confirmed by the quantitative evaluation, in which we outperform our competitors across all cross-reenactment metrics (Table 1), including both user studies.

Cross-reenactment

Method	FID↓	CSIM↑	IQA↑
Base w/ bicubic	51.4	0.67	35.1
HiFaceGAN	49.4	0.65	43.9
Ours	39.2	0.67	49.3

Table 2: Quantitative results on the FFHQ dataset in the cross-reenactment mode at 1024×1024 resolution. Besides the standard cross-reenactment metrics, we additionally perform an image quality assessment (IQA, higher the better). Our super-resolution method improves the resulting image quality compared to the base model with bicubic upsampling and the super-resolution baseline (HiFaceGAN), as seen from the FID and IQA metrics. At the same time, we preserve the source image appearance, which results in the same CSIM as the base model.

4.4 Self-reenactment evaluation

We use the same pre-trained models for the self-reenactment experiments as for the cross reenactment and evaluate them on the samples from the VoxCeleb2 and VoxCeleb2HQ evaluation sets. In addition, we use the following standard metrics to measure the difference between the synthesized and ground-truth images: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [35], and the Learned Perceptual Image Patch Similarity (LPIPS) [40].

We notice that qualitatively we achieve similar performance to the competitors, especially in the face and hair regions (Figures 3-4, third row). To quantitatively verify that, we have conducted an evaluation using masked data. The masks include the face, ears, and hair regions and are applied to both the target and the predicted images before calculating the metrics. In this scenario, we achieve comparable performance to the baseline methods (Table 1) but have an inferior performance when the unmasked (raw) images are used.

This difference could be caused, among other reasons, by the lack of shoulders motion modeling in our method. It results in the misalignment between our predictions and ground truth in the corresponding regions. We further discuss this issue in the limitations section. Also, our method's high degree of disentanglement between motion and appearance descriptors prevents it from leaking the appearance data directly from the driver, which generally contributes to the reduced performance in self-reenactment.

4.5 High-resolution evaluation

We evaluate high-resolution synthesis only in cross-reenactment mode since data for the self-reenactment scenario is missing. We use subsets of a filtered FFHQ dataset for training and evaluation. We train both our and the baseline super-resolution approaches using an output of a pre-trained base model $\mathbf{G}_{\text{base}}$ as input and by sampling two random augmented versions of the training image as a source and a driver. We use random crops and rotations since other augmentations could change person-specific traits (e.g. head width).

We compare against two baselines. First, we consider bicubic upsampling of the output of the base model, and second, we evaluate

Figure 5: A qualitative comparison of different super-resolution methods applied to the output of our base model. While performing better than a baseline bicubic upsampling, we can see that the state-of-the-art super-resolution method (HiFaceGAN) cannot achieve the same level of high-frequency details fidelity as our approach. Digital zoom-in is recommended.

a state-of-the-art face super-resolution system (HiFaceGAN) [37]. The results are presented in Figure 5, and Table 2. In the quantitative comparison, we use an additional image quality assessment metric (IQA) [28] to measure the resulting image quality. Our method outperforms its competitors both qualitatively and quantitatively by generating more high-frequency details and, at the same time, preserving the identity of the source image.

Finally, in Figure 6 we show the results for the distillation of our base and high-resolution models into a small student network designed to work for a limited number of avatars. The architecture we chose for the distillation achieves 130 frames per second on the NVIDIA RTX 3090 graphics card in the FP16 mode. The total model size for the student containing 100 avatars is 800 megabytes. This model can closely match the performance of the teacher model. It thus achieves a PSNR of 23.14 and LPIPS of 0.208 (w.r.t. the teacher model) averaged across all avatars.

Figure 6: Results of the distilled version of our system trained for 100 avatars. It closely matches the prediction of the teacher model while being approximately ten times faster at the inference, achieving up to 130 FPS on a modern GPU.

4.6 Ablation study

We conducted an extensive ablation study to evaluate the contributions of individual components within our method. Therefore, we evaluate the importance of the proposed cycle consistency losses for the base and high-resolution models. The qualitative results are shown in Figure 7. Overall, both losses substantially improve the disentanglement between the motion and appearance. The quantitative evaluation confirms this: the base model without $\mathcal{L}_{\text{cos}}$ achieves an FID of 34.8, compared to the final 28.6, and the high-resolution model without cycle losses has an FID of 39.6, compared to the final FID of 39.2. We also provide an in-depth evaluation of the architectural choices in the supplementary materials.

Figure 7: Ablation study. Both contrastive loss $\mathcal{L}_{\text{cos}}$ and unsupervised super-resolution losses $\mathcal{L}^{\text{c}}_{\text{adv}}$ and $\mathcal{L}^{\text{c}}_{\text{cyc}}$ (denoted as $\mathcal{L}^{\text{c}}_{*}$) improve the performance of our method in the cross-driving scenario.

5 CONCLUSION

We have presented a new approach for synthesizing high-resolution neural avatars. To the best of our knowledge, this approach is the first to achieve megapixel resolution. We have also explored a possible application of the proposed method in practice, which involves locking the identities of the avatars by training a dedicated student network. Using the student network also increases the

Figure 8: The limitations of our method include the inability to model large head rotations, which stems from the near frontal views distribution in the training data (1st example), and the lack of shoulders motion modeling (2nd example).

rendering speed while achieving similar quality of renders to our full one-shot model.

Two main limitations of our system stem from the properties of our training set. First, both the VoxCeleb2 and the FFHQ datasets that we use for training tend to have near frontal views, which degrades the quality of rendering for strongly non-frontal head poses (Figure 8). Secondly, as only static views are available at high resolution, a certain amount of temporal flicker is present in our results (see supplementary video). Ideally, this needs to be tackled with special losses or architectural choices. Lastly, our system lacks the modeling of shoulders motion. Addressing the issues mentioned above remains our future work.

ACKNOWLEDGEMENTS

We sincerely thank Michail Christos Doukas for providing us inference results of HeadGAN [8] system and Ting-Chun Wang for providing us inference results of Face-V2V [34] system. We also thank Roman Suvorov for their comments and suggestions regarding the text contents and clarity, as well as Julia Churkina for helping us with proof-reading. The computational resources for this work were mainly provided by Samsung ML Platform.

REFERENCES

[1] Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *SIGGRAPH '99*.

[2] Adrian Bulat and Georgios Tzimiropoulos. 2017. How Far are We from Solving the 2D & 3D Face Alignment Problem? (and a Dataset of 230,000 3D Facial Landmarks). *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 1021–1030.

[3] Egor Burkov, I. Pasechnik, Artur Grigorev, and Victor S. Lempitsky. 2020. Neural Head Reenactment with Latent Pose Descriptors. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 13783–13792.

[4] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. VoxCeleb2: Deep Speaker Recognition. In *INTERSPEECH*.

[5] Kevin Cortacero, Tobias Fischer, and Yiannis Demiris. 2019. RT-BENE: A Dataset and Baselines for Real-Time Blink Estimation in Natural Environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

[7] Jiankang Deng, J. Guo, Evangelos Ververas, Irene Kotsia, Stefanos Zafeiriou, and InsightFace FaceSoft. 2020. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 5202–5211.

[8] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. 2021. HeadGAN: One-shot Neural Head Synthesis and Editing. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*.

[9] Tobias Fischer, Hyung Jin Chang, and Y. Demiris. 2018. RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments. In *ECCV*.

[10] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. 2021. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8645–8654.

[11] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, M. Wang, and Liang Lin. 2019. Graphonomy: Universal Human Parsing via Graph Transfer Learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 7442–7451.

[12] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. 2020. MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets. In *AAAI*.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*.

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*.

[15] Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4396–4405.

[16] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson W.H. Lau. 2022. MODNet: Real-Time Trimap-Free Portrait Matting via Objective Decomposition. In *AAAI*.

[17] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. 2018. Deep video portraits. *ACM Transactions on Graphics (TOG)* 37 (2018), 1 – 14.

[18] Stephen Lombardi, Jason M. Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)* 37 (2018), 1 – 13.

[19] Stephen Lombardi, Tomas Simon, Jason M. Saragih, Gabriel Schwartz, Andreas M. Lehrmann, and Yaser Sheikh. 2019. Neural volumes. *ACM Transactions on Graphics (TOG)* 38 (2019), 1 – 14.

[20] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

[21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

[22] Keunhong Park, U. Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable Neural Radiance Fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*.

[23] Keunhong Park, U. Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. 2021. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *ArXiv*.

[24] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep Face Recognition. In *BMVC*.

[25] Aliaksandr Siarohin, Stéphane Lathuilière, S. Tulyakov, Elisa Ricci, and N. Sebe. 2019. Animating Arbitrary Objects via Deep Motion Transfer. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 2372–2381.

[26] Aliaksandr Siarohin, Stéphane Lathuilière, S. Tulyakov, Elisa Ricci, and N. Sebe. 2019. First Order Motion Model for Image Animation. *ArXiv* abs/2003.00196 (2019).

[27] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2015).

[28] Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. 2020. Blindly Assess Image Quality in the Wild Guided by a Self-Adaptive Hyper Network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[29] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. 2021. Resolution-robust Large Mask Inpainting with Fourier Convolutions. *arXiv preprint arXiv:2109.07161* (2021).

[30] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2019. Face2Face: real-time face capture and reenactment of RGB videos. *ArXiv* abs/2007.14808 (2019).

[31] H. Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jin Zhou, and Wenyu Liu. 2018. CosFace: Large Margin Cosine Loss for Deep Face Recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 5265–5274.

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

[33] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[34] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. 2021. One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[35] Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13 (2004), 600–612.

[36] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. 2021. BANMo: Building Animatable 3D Neural Models from Many Casual Videos. *ArXiv*.

[37] Lingbo Yang, C. Liu, P. Wang, Shanshe Wang, P. Ren, Siwei Ma, and W. Gao. 2020. HiFaceGAN: Face Renovation via Collaborative Suppression and Replenishment. *Proceedings of the 28th ACM International Conference on Multimedia* (2020).

[38] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor S. Lempitsky. 2020. Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars. In *ECCV*.

[39] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor S. Lempitsky. 2019. Few-Shot Adversarial Learning of Realistic Neural Talking Head Models. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

[40] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 586–595.

[41] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and S. Li. 2017. S3FD: Single Shot Scale-Invariant Face Detector. *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 192–201.

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 2242–2251.

A NETWORK ARCHITECTURES

Here we describe the architectures of our networks, conduct an in-depth evaluation of the architectural choices and provide details about preprocessing of the dataset.

A.1 Base model

In the architecture of our base model, we replace all BatchNorms with GroupNorms, and all convolutional layers, except the first and the last ones, are used with weight standardization.

Appearance encoder $\mathbf{E_{app}}$. The network consists of two parts. The first part produces a 4D tensor of volumetric features v_s that represent the person's appearance from the source image. It includes several residual blocks followed by average pooling. We reshape the resulting 2D features to 3D features and then apply several 3D residual blocks to compute the final volumetric representation. The scheme shown in Figure 9 (a).

The second part produces a global descriptor $\mathbf{e}_s$ that helps retain the appearance of the output image. We use a ResNet-50 architecture with custom residual blocks. The architecture of our residual block can be seen in Figure 11 (c), where n denotes the dimension of a convolutional layer (either 2D or 3D) and $\mathbf{x}$ denotes the number of output channels.

Motion encoder $\mathbf{E_{mtn}}$. We use two separate ResNet-18 networks as encoders to separately predict the head pose and expression vector. The head pose prediction network is pre-trained, while the expression prediction network is trained from scratch.

Warping generators $\mathbf{W}_{s\rightarrow}$ and $\mathbf{W}_{\rightarrow d}$. When both source and driver tuples $(\mathbf{R}_{s/d}, \mathbf{t}_{s/d}, \mathbf{z}_{s/d}, \mathbf{e}_s)$ are predicted, we can produce 3D warping $\mathbf{w}_{s\rightarrow}$ and $\mathbf{w}_{\rightarrow d}$. Both warpings consist of two parts: one in charge of rotation and translation ($\mathbf{w}^{\text{rt}}_{\dots}$) and another one in charge of emotion changing ($\mathbf{w}^{\text{em}}_{\dots}$).

To get the first part: for $\mathbf{w}^{\text{rt}}_{\rightarrow d}$ we multiply identity grid on transformation matrix and for $\mathbf{w}^{\text{rt}}_{s\rightarrow}$ we multiply identity grid on inversed transformation matrix.

To get $\mathbf{w}^{\text{em}}_{s\rightarrow}$ and $\mathbf{w}^{\text{em}}_{\rightarrow d}$ we use two separate warping generators (see Figure 9 (b)) with the same architecture contain several 3D residual blocks where all GroupNorms changed on Adaptive GroupNorms (marked as ResBlock3D* on the scheme), whereas inputs we use sums $\mathbf{z}_s + \mathbf{e}_s$ and $\mathbf{z}_d + \mathbf{e}_s$ respectively. To generate adaptive parameters, we multiply the foregoing sums and additionally learned matrices for each pair of parameters.

3D convolutional network $\mathbf{G_{3D}}$. Next, we process volumetric representation after the first warping to get canonical volume where source motion removed from the appearance features. For this, we apply Unet-like architecture with several downsample units consists of 3D residual block and downsample operation, followed by the same number of upsample units consists of 3D residual block and upsample operation. The scheme shown in Figure 9 (c). Sample(z, x, y) mean sample operation that changes depth, height, width in z, x, y times respectively. For example, z=1/2 means downsample along depth dimension in 2 times and z=2 means upsample along depth dimension in 2 times.

2D convolutional network $\mathbf{G_{2D}}$. Finally, to predict an output image from a processed volume, we first use orthographically projection $\mathcal{P}$ that is consists of reshape operation and 1x1 convolution. While more complex projection operators could be used (like volumetric ray marching), we found such simple approach is sufficient for our applications, the same way as it has been done in [32]. Then we utilize the network includes 8 residual blocks on the same resolution and number of feature maps, then gradually apply units contain upsampling and residual blocks with successively decreasing number of output channels. The scheme shown in Figure 9 (d).

A.2 High-resolution model

High-resolution model contain 2 parts: encoder and decoder. Both of them you can see on the scheme shown in Figure 11. **Encoder**, that takes $\hat{\mathbf{x}}_{s->d}$ as an input, contain just conv layer followed by 2 residual blocks and produce 3D feature tensor $\mathrm{f}_{s->d}$. **Decoder**, that takes output features $\mathrm{f}_{s->d}$ and produce hi-resolution version of input $\hat{\mathbf{x}}^{\text{HR}}_{s->d}$, recalls 2D convolutional network from the base model, it also includes 8 residual blocks on the same resolution and number of feature maps, followed by two upsampling with residual blocks and three residual blocks on high-resolution.

A.3 Student model

The encoder of a student model is ResNet18, and the generator consists of residual blocks with SPADE normalization layers, in each SPADE block a tensor used for normalization is fixed for a specific avatar. During the forward pass we select which tensor to use in normalization layer to switch between predefined avatars. Using such procedure during the training, we force our model to store all the identity-specific information into SPADE blocks. Also, to compress the final model we tweak a size of spatial dimension of normalization tensors (which dominate the size of the whole model) in SPADE blocks: by default these tensors must be of the same shape as a corresponding input feature map, instead, we compress them spatially and use bilinear upsampling to output the feature map of the right size. More precisely, we bound the resolution of an inner identity tensor by 64.

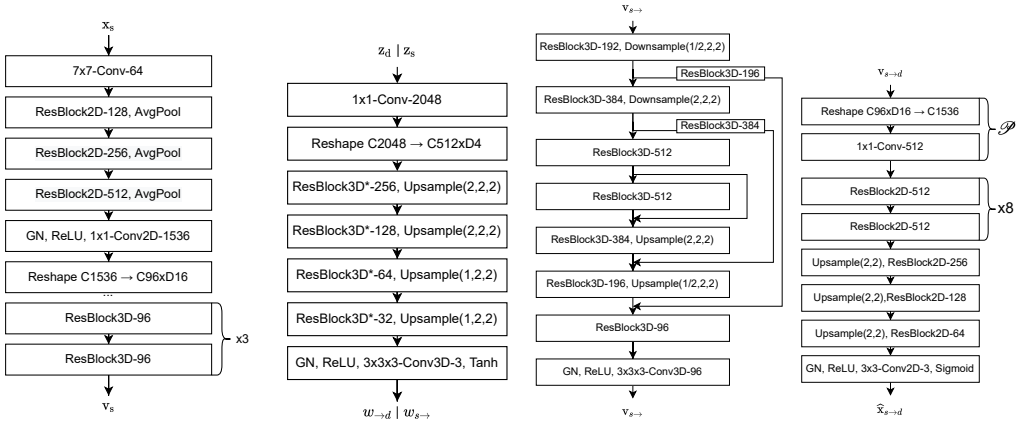
B ADDITIONAL INFORMATION

B.1 Training details

$\mathbf{G_{base}}$ and $\mathbf{G_{HR}}$. As augmentation for both source and target images, we use color jitter and random flip. As for driving image, before sending it to $\mathbf{E}_{\text{mtn}}$ we do a center crop around the face of a person. Next, we augment it using a random warping based on thin-plate-splines, which severely degrades the shape of the facial features, yet keeps the expression intact (ex., it cannot close or open eyes or change the eyes' direction). Finally, we apply a severe color jitter.

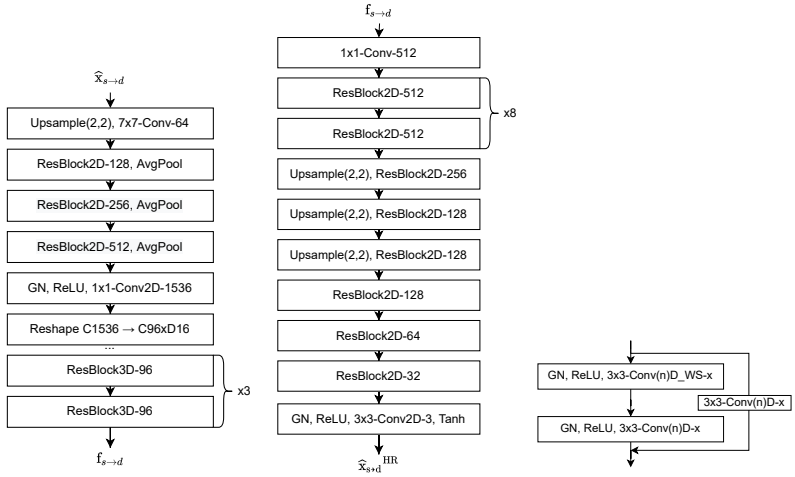
For $\mathcal{L}_{\text{GAN}}$ loss we use multi-resolution patchGAN where the discriminator produces the patch-level prediction. We apply spectral normalization for both $\mathbf{G}_{\text{base}}$ and $\mathbf{G}_{\text{HR}}$.

For AdamW optimizer we used the following parameters: betas=(0.5, 0.999), eps=1e-8, weight decay=1e-2 for both $\mathbf{G}_{\text{base}}$ and $\mathbf{G}_{\text{HR}}$ and correspond discriminator.



E_{app} (a) **$W_{\to d} \mid W_{s\to}$ (b)** **G_{3D} (c)** **G_{2D} (d)**

Figure 9: Architectures of components of G_{base}.



Encoder (a) **Decoder (b)** **Resblock (c)**

Figure 10: Architectures of components of G_{enh}.

Student

The student model was trained to predict the corresponding prediction of the teacher model for a fixed set of identities. We used a standard set of losses for such setup, specifically, adversarial and three kinds of perceptual losses. Adversarial training was done with multiscale discriminator on four resolutions. Perceptual losses are the same that were used to train teacher model: standard VGG19 loss, gaze loss and VGG Face loss. Additionally, to check how student model handle self-reenactment mode, we train separate student model on 10 avatars, using persons from 10 random test videos. Student model achieved PSNR of 19.25 and SSIM of 0.682 (while teacher model achieved 21.34 and 0.768 correspondingly). You can see an example in Figure 17.

B.2 Two stage training

Initially, we have evaluated some of the feasible configurations for the end-to-end training. First of all, end-to-end training with the full enhancer network or even a single decoder layer at 1024x1024 resolution would not fit into the memory of our available GPUs. We, therefore, tried freezing a pre-trained encoder and fine-tuning a decoder with an additional upsampling block, combining both of the objectives and with some weighting coefficient. We have observed a significant decrease in the quality of the results, compared to a separately trained network, across three different weights. Since it is effectively doing super-resolution without high-resolution conditioning, maintaining its high capacity is crucial for the network to generate the missing high-frequency details. You can see some comparison in Figure 9.

B.3 Datasets preprocessing

We obtain the VoxCeleb2HQ dataset by first downloading the original videos from the VoxCeleb2 [4] dataset. These videos are processed using an off-the-shelf face [41] and keypoints [2] detectors and cropped frame-by-frame around the head regions. Then, the obtained cropped frames are first filtered by their resolution, to

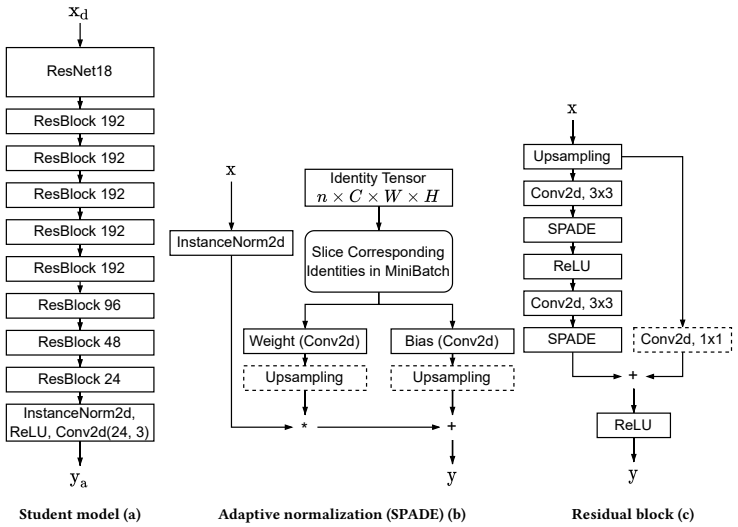


Figure 11: Architectures of components of the student model. Dashed lines correspond to the optional blocks (i.e. used only if channel/resolution configuration needs to apply some transform, either upsampling or change the number of channels). Each SPADE tensor shape $W \times H$ is at most 64×64.

exclude all crops that are smaller than 512×512. We call this process a bitrade filtering. Then, we additionally rank the frames from the remaining videos by their image quality assessment (IQA) scores, calculated using a pre-trained system [28]. We then remove the bottom 50% of the videos using the mean IQA score across its frames, thus arriving at the 15,000 videos that we use for training.

B.4 Evaluation of the architectural choices

In addition to the ablations we describe in the main part of the paper, we decided to conduct a series of additional experiments to evaluate the impact of the key features in our method. We demonstrate results of an additional ablation study in Figure 12. The following parts were eliminated separately: (a) architecture without encoder $\mathbf{e}_s$, (b) the warping for source image $\mathbf{w}_{s\rightarrow}$, (c) driver augmentation, (d) added block to predict background directly with person appearance, (e) base model. We show that $\mathbf{e}_s$ helps to preserve identity information, especially on tight turns. Without warping generator $\mathbf{w}_{s\rightarrow}$ the preservation of whole structure of the shoulders and head worsen and artifacts appeared on ears. If turn off driver augmentation, apparently the model shows significant worsening of results in terms of identity preservation (see the eyes and ears zones, and artifacts on the cap and temples). Additionally, we train our model to predict background and person together. The preservation of the identity dropped as long as the whole image quality. Mainly because the capacity spent on the background modeling.

B.5 Gaze loss

To get more natural facial appearance, we put into operation specialized gaze loss based on gaze and blink estimation models. Our model was trained to distill a state-of-the-art gaze detection system (RT-GENE) [9] and blink estimation model (RT-BENE). [5]. We distilled two systems in one model with two heads with the common backbone, one to predict gaze direction and another one to predict blink. First, we infer both models on 60k random frames from our dataset. We did this in order to extract the maximum information from the images of the eyes. As a backbone for our model we used VGG-16 that takes one image of the eye (either left or right) and predict latent vector with size 256, next we sum both vector to get bound representation of eyes. We also derive features from 2D keypoints, for this we use a simple network consists of 3 liner layers with ReLU activations that produce latent vector with size 64. Next, we utilize 2 separate heads, both contain only 2 liners layers with ReLU activations. For the gaze prediction head we use as an input sum of eye vectors concatenated with keypoint vector and for blink prediction just sum of eye vectors.

We train this model for 60 epoches with batch size equal to 64. We use Adam optimizer with initial learning rate equal to 0.8e-3, betas=(0.9, 0.999), eps=1e-08, weight decay=0 and one cycle learning rate schedule with steps per epoch equal number of batches in epoch and pct start=0.1. We use MAE and MSE losses with $w_{\text{MAE}} = 15$ and $w_{\text{MSE}} = 10$, we treat predictions from RT-GENE and RT-BENE as ground truth.

B.6 Explicit control of the pose

Our system allows some explicit control of a human pose on an output image. First, we can either preserve scale of the source image, that could be utilized in video conference, or use scale and translation (*s&t*) from the driving image to fully mimic the driver (Figure 13). Despite the fact that we didn't pay any attention to disentangle expression and head rotation, we found that we can both make formalization (Figure 14) and apply head rotation from frontal pose on moderate angels, we found that it works at least for 15°angles (Figure 15).

Source | Driver | w/o e_s (a) | w/o $w_{s\rightarrow}$ (b) | w/o augs (c) | w/ BG (d) | Ours

Figure 12: Additional ablation study. We qualitatively evaluate the individual components of our base model (last column). We observe the positive influence of crucial part of our method. The details of the evaluation described in Section 2.3.

Source | Driver | Source *s&t* | Driver *s&t*

Figure 13: Results with different scales and translations

Source | Reconstructed | Frontalized

Figure 14: Result of frontalization

C ADDITIONAL RESULTS

We demonstrate the comparison of our method for both cross- and self-reenactment in Figure 18 for 256×256 resolution and in Figure 19 for 512×512. Also, we show qualitative comparison in cross-reenactment scenario for 1024×1024 resolution in Figure 20.

Moreover, we attach a few demonstration videos for one megapixel resolution and video comparison for cross-reenactment 256×256 and self-reenactment 512×512 scenarios. We strongly encourage reader to check this video.

One of the interesting points is that the model learns meaningful features in volume tensor, that encodes the geometry of the give source to $\mathbf{v}_s$ with the shape $96 \times 16 \times 64 \times 64$ and attach the video of animation this volumetric tensor in supplementary files.

Source | Roll +15° | Roll -15° | Roll +15° | Roll -15° | Pitch +15° | Pitch -15°

Figure 15: Result of the explicit head rotation.

Source | Driver | One stage, fine-tuned | Final

Figure 16: A qualitative comparison of one stage training with fine-tuned decoder and our final two stage training. Pay special attention to the area around the eyes, glasses, teeth, hair and skin, where the difference between the two approaches is most noticeable.

Source | Driver | Teacher | Student

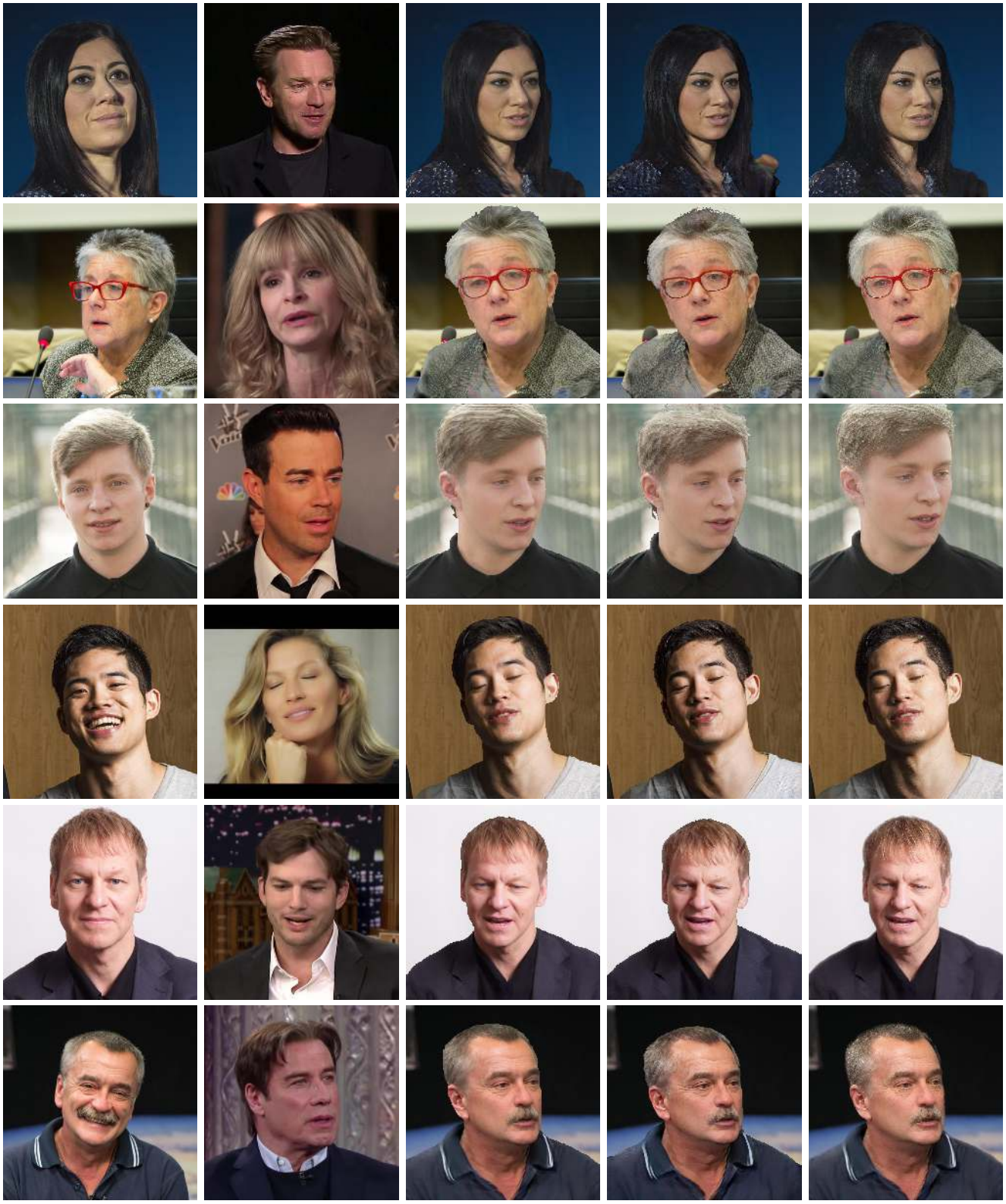
Figure 17: Result of student model in self-reenactment mode.

Source **Driver** **FOMM** [26] **HeadGAN** [8] **Ours**

Figure 18: A qualitative comparison of head avatar systems in cross-reenactment scenario (top four rows) and self-reenactment scenario (bottom two rows) at 256×256 resolution.

Source **Driver** **Face-V2V** [34] **Ours**

Figure 19: A qualitative comparison of head avatar systems in cross-reenactment scenario (top three rows) and self-reenactment scenario (bottom two rows) at 512px resolution.



Source Driver Ours (base w/ bicubic) HiFaceGAN [37] Ours (HR)

Figure 20: A qualitative comparison of different super-resolution methods applied to the output of our base model. While performing better than a baseline bicubic upsampling, we can see that the state-of-the-art super-resolution method (HiFaceGAN) cannot achieve the same level of high-frequency details fidelity as our approach. Digital zoom-in is recommended.