Skolkovo Institute of Science and Technology

Synthesis of Human Face and Body Images via Generative Adversarial Networks

Doctoral Thesis

by

Egor Zakharov

DOCTORAL PROGRAM IN COMPUTATIONAL AND DATA SCIENCE AND ENGINEERING

Supervisor
Victor Lempitsky

Moscow - 2023

I hereby declare that the work presented in this thesis was carried out by myself at Skolkovo Institute of Science and Technology, Moscow, except where due acknowledgment is made and has not been submitted for any other degree.

Candidate (Egor Zakharov)

Supervisor (Dr. Victor Lempitsky)

Abstract

This thesis tackles the problem of a human face and body image synthesis. The generative models with such capabilities are referred to in the literature as *avatars* and have become a key component of modern telepresence and special effects systems. A few years ago, the creation and realistic rendering of avatars was a challenging and resource-hungry process that required specialized capture setups, such as light stages that can cost millions of dollars and hours of manual post-processing time of computer graphics artists. Nowadays, head and facial capture and subsequent realistic reenactment can be done using just a collection of images produced via a smartphone. In this manuscript, we present some of the methods that contributed to the emergence of these modern systems. The leitmotif of our work is in leveraging the deep neural networks and their training techniques, such as Generative Adversarial Networks (GANs), perceptual losses, and meta-learning, to achieve high levels of avatars' personalization and rendering quality.

First, we combined the widely spread GAN-based and perceptual losses in a single system, which showed improvements in the facial editing task. Then, we tackled the problem of full-body avatar synthesis using a novel architecture based on neural networks. We integrated classical computer graphics and neural-based approaches to achieve high robustness of the learned avatars w.r.t. new camera poses unseen during training. However, the resulting model still worked in a so-called multi-shot learning scenario, i.e. when each avatar system was trained to model a particular individual given a large amount of data depicting their appearance and motion. In the follow-up work, we were among the first to introduce large-scale training into the avatar creation process to achieve the realistic synthesis of human heads in a few-shot scenario, i.e. given just a handful of images per person. To accomplish that, we tackled this problem via meta-learning to improve the personalization of the avatars and GAN-based training to ensure the high realism of renders. Next, to facilitate practical applications for such systems, we developed a new model that allowed the creation and rendering of realistic head avatars even on mobile devices. In one of the most recent works, we extended the existing capabilities for cross-subject reenactment by introducing a novel way of learning implicit latent representations of human facial expressions. This new approach allowed us to achieve state-of-the-art cross-subject reenactment and the degrees of fidelity in motion transfer that were unseen in prior works. We also proposed a novel method for scaling the few-shot head avatar systems to megapixel resolution. Lastly, we have explored the challenging task of one-shot rigged head mesh reconstruction and were the first to propose a method capable of reconstruction, subsequent animation, and neural rendering of the human heads using just a single image.

List of Publications

1. Diana Sungatullina*, Egor Zakharov*, Dmitry Ulyanov, Victor S. Lempitsky. **"Image Manipulation with Perceptual Discriminators"**. *15th European Conference on Computer Vision (ECCV), 2018, *denotes equal contribution.*

2. Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, I. Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, Victor S. Lempitsky. **"Textured Neural Avatars"**. *2019 IEEE/CVF Conference on Computer Vision, Pattern Recognition (CVPR).*

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

4. Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, Victor S. Lempitsky. **"Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars"**. *16th European Conference on Computer Vision (ECCV), 2020.*

5. Nikita Drobyshev, Evgeny Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor S. Lempitsky, Egor Zakharov. **"MegaPortraits: One-shot Megapixel Neural Head Avatars"**. *30th ACM International Conference on Multimedia (ACMMM), 2022.*

6. Taras Khakhulin, Vanessa Sklyarova, Victor S. Lempitsky, Egor Zakharov. **"Realistic One-shot Mesh-based Head Avatars"**. *17th European Conference on Computer Vision (ECCV), 2022.*

* denotes joint first authorship. The author of the thesis made the following contributions to the papers where he is not the single first author:

- **Image Manipulation with Perceptual Discriminators**: co-developing the training method, preparing the experimental codebase, conducting the part of the main experiments and contributing to paper writing and figure preparation.

- **Textured Neural Avatars**: preparing and evaluating the baseline methods, major contribution to paper writing and the preparation of the figures, conducting initial experiments with generative adversarial networks.

- **MegaPortraits: One-shot Megapixel Neural Head Avatars** and **Realistic One-shot Mesh-based Head Avatars**: overall supervision of the projects leading to the publication, suggesting and developing ideas behind the systems, preparing the codebase, conducting part of the experiments, writing a large portion of the text of the paper, preparing the figures.

Acknowledgements

I want to thank my scientific advisor, Victor Lempitsky, sincerely. Their role in my professional growth is immeasurable, and I am happy to have had an opportunity to work together for the past six years during both my M.Sc. and Ph.D.

I thank Skoltech and wish the university a bright future, as it allowed me to become a part of a world-class research group. I have also been a full-time employee at Samsung AI Center in Moscow since the start of my Ph.D. in 2018. I thank Samsung for allowing me to formulate and execute my research agenda by giving me access to computational resources and the ability to work with amazingly talented colleagues.

I especially want to thank Dmitry Ulyanov, whose mentorship and scientific brilliance have inspired me and helped kick-start my career.

Last but not least, I thank my wife, Irina, who has been incredibly supportive and shared all the highs and lows with me. I am also grateful to my parents, especially my mother, for providing me with the best education opportunities.

Contents

List of Abbreviations

3D Three dimensional

AdaIN Adaptive Instance Normalization Huang and Belongie [2017]

CNN Convolutional Neural Network LeCun et al. [1998]

CPU Central processing unit

DensePose Dense Human Pose Estimation, a full-body pose estimation system Güler et al. [2018]

GAN Generative Adversarial Networks, a generative model Goodfellow et al. [2014a]

GPU Graphics processing unit

MLP Multi-Layer Perceptron Rumelhart et al. [1986b]

PCA Principal component analysis Pearson [1901]

ReLU Rectified Linear Unit Maas [2013]

ResNet Residual Network, a neural network architecture He et al. [2016a]

RGB Red-green-blue, channels in a digital representation of an image

VAE Variational Autoencoder, a generative model Kingma and Welling [2014]

VGG A neural network architecture, proposed by Visual Geometry Group, University of Oxford Simonyan and Zisserman [2014]

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is based on the collection of publications on the human face and body synthesis. The unifying factor between these works is the usage of generative adversarial networks, which help achieve a high degree of realism in the synthesized images, as well as extensive usage of large-scale datasets for the training of such networks, which facilitates both the creation of the avatars and their rendering.

1.1 Motivation

The synthesis of realistic human images has historically been one of the most challenging problems in computer graphics. Its complexity largely stems from the effect known as *uncanny valley* Mori [1970], which manifests in a negative emotional response of humans to the depictions of anthropomorphic subjects when the latter reach a certain level of human-likeness or realism. This effect is significantly exacerbated when the respondents are shown moving or talking subjects, which constitutes a challenging obstacle to generating realistic and animated depictions of humans.

Initial success in synthesizing realistic images of human subjects was demonstrated Blanz and Vetter [1999], Debevec et al. [2000], Kanade et al. [1997] using computer graphics techniques such as photogrammetry Kanade et al. [1997], structured light Debevec et al. [2000] or laser 3D scanning Blanz and Vetter [1999], Figure 1.1. These methods reconstruct the subject in the form of a polygonal mesh and a corresponding colored texture or albedo. Such representation allows the synthesis of novel images by rendering the mesh from different angles and in various lighting conditions. The obtained reconstructions could be subsequently animated either completely manually or by refining the data gathered from the motion capture systems Gavrila and Davis [1996]. This pipeline can achieve state-of-the-art results even to this day and is widely used in the

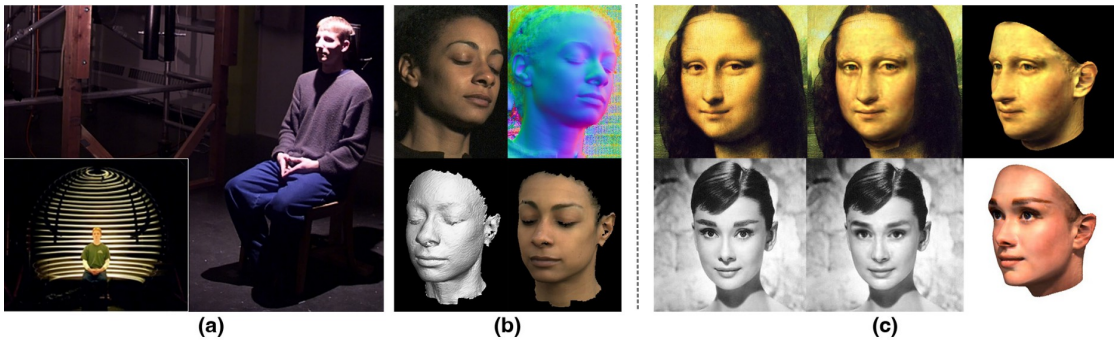


FIGURE 1.1: A brief overview of the two influential classical human modeling methods. The *Light Stage* Debevec et al. [2000] is the first version of the structured light 3D scanner (a), in which the facial shape and appearance of the subject (b) are reconstructed using pre-determined light patterns. While achieving high realism of the facial reconstructions, the capture process requires highly specialized equipment. In parallel, statistical mesh modeling Blanz and Vetter [1999] methods were developed that achieved lower accuracy and personalization, but were capable of facial reconstruction using only a single image (c).

film and computer games industries. However, such a process requires access to highly specialized equipment for appearance and motion capture, as well as manual input from professional artists who refine the captured data to create the photorealistic renders of the characters we are used to seeing on the big screen.

It is worth mentioning that the refinement of the motion capture data is rarely carried out at the level of individual mesh vertices. Instead, the lower dimensional representations, such as keypoints or blendshapes, are introduced by *rigging* the underlying mesh model. Such a rigging process can be either manual or automatic. The latter relies on the statistical mesh modeling systems Blanz and Vetter [1999], Li et al. [2017], Loper et al. [2015], Pavlakos et al. [2019], Paysan et al. [2009], typically referred to as *parametric models*. One of their key ideas was primarily inspired by the influential work on the Eigenfaces Sirovich and Kirby [1987], in which a principal component analysis (PCA) Pearson [1901] was performed on a dataset of the greyscale facial images. It allows each facial image to be represented as a linear combination of eigenvectors, or the so-called "eigenfaces", drastically reducing their dimensionality. The early parametric mesh models Blanz and Vetter [1999] leveraged this idea by applying PCA to the dataset of 3D scanned human faces, obtaining the basis vectors both for the mesh vertices and the textures, which allows introducing rigging for arbitrary 3D scans without any manual input. The dimensionality reduction property of the PCA also helps estimate these parametric models even in ill-posed scenarios, such as reconstruction from a monocular video obtained with an uncalibrated camera or even a single image Feng et al. [2018], Thies et al. [2016b]. These models have also been introduced for the full body Loper et al. [2015], head Li et al. [2017], and hands Pavlakos et al. [2019], effectively covering the majority of the human body motion.

While such statistical approaches were quite successful in representing the motion of a subject, in many regards, the obtained quality of 3D reconstructions and rendering was lacking. First,

obtaining faithful 3D scans of the mouth cavity and hair has proven challenging, and therefore they were not present in the existing models Blanz and Vetter [1999], Li et al. [2017]. A similar problem occurs in full-body reconstructions with clothing representation Loper et al. [2015], Pavlakos et al. [2019] since its diversity and non-rigidness prevent dimensionality compression using linear methods. Second, creating photorealistic renders typically requires a high number of parameters to imprint person-specific high-frequency details onto the mesh geometry and texture. However, due to their low dimensionality, parametric models are only sufficient to represent and realistically render facial regions. Lastly, the PCA-based dimensionality reduction inevitably leads to the loss of capability to represent subtle motions such as wrinkles and detailed mimics. These motions are crucial for the truthful rendering of the humans that avoids the uncanny valley effect. To this end, realistic synthesis of the face and body images was only made possible by replacing the classical rendering pipeline with the so-called *neural rendering*, which heavily relied on neural networks to produce realistic images and faithful representation of the motion.

For the past decade, deep artificial neural networks have been widely used in various areas of computer vision. However, their initial groundbreaking results were achieved by addressing the problems related to discriminative modeling. The pioneering works Rumelhart et al. [1986a] first popularized neural networks in their "shallow" version by proposing a highly effective error backpropagation algorithm to train them. This algorithm was later applied LeCun et al. [1998] to the so-called *convolutional* neural networks (CNNs) and achieved impressive results in classifying image-like data. However, the beginning of their widespread usage of neural networks corresponds to the "deep learning revolution", a term coined in a similarly named book Sejnowski [2018]. The advancements in graphical processing units (GPUs) have led to the development of the GPU-based backpropagation frameworks Krizhevsky et al. [2012], which leveraged the inherent data parallelism in the convolutional operations to speed up their optimization process. This made it feasible to use *deep* neural networks, which had a substantially higher complexity and the number of parameters than their shallow counterparts. Concurrently, large-scale classification datasets with human-made annotations started to emerge Everingham et al. [2009], Russakovsky et al. [2015], which deep CNNs were empirically shown to benefit from. In the end, in a milestone achievement, deep neural networks eventually outperformed Simonyan and Zisserman [2015b] the classical computer vision methods in the image classification task.

Deep convolutional networks were shown to be highly effective not only in discriminative tasks but also in generative modeling. Image colorization Zhang et al. [2016b], stylization Gatys et al. [2016], and super-resolution Johnson et al. [2016b] tasks were all shown to benefit from using CNNs. The generation of realistic images soon followed with the groundbreaking work by Goodfellow et al. Goodfellow et al. [2014a], in which generative adversarial networks (GANs) were introduced. This method quickly overtook the abovementioned areas, as the introduction of adversarial training was shown Isola et al. [2017], Ledig et al. [2017a], Zhu et al. [2017c] to improve results in all these tasks. The key advantage of this method, compared to the competitors

such as variational autoencoders (VAEs) Kingma and Welling [2014], was the perceived realism of the obtained images. It was achieved by hallucinating convincing high-frequency details while maintaining global semantic consistency.

These neural methods were also successfully applied to the problems of human image synthesis, ranging from semantic image editing Choi et al. [2018], Zhu et al. [2017a] to novel view synthesis Thies et al. [2019a] and facial animation Kim et al. [2018b]. In Chapter 2, we introduce one of such semantic image editing systems, in which the unsupervised GAN-based training procedure was improved using networks pre-trained on large-scale datasets. While being effective for unsupervised learning, we did not employ this training method in our follow-up works. It was shown Chan et al. [2018], Wang et al. [2018b] that human avatars can be learned via supervised training using de-personalized motion representations, such as keypoints. This approach has proven its high efficiency compared to unsupervised training, and we used it for the majority of our follow-up work.

Some of the abovementioned avatar systems combine classical data structures and their rendering pipelines from computer graphics with neural networks. However, in Chapters 3-4, we show that realistic synthesis of such images could be achieved without explicitly defined 3D models. In Chapter 3, we describe a system which is utilizing a mapping from the image space to texture space, directly estimated from the input image without any underlying 3D reconstruction process Güler et al. [2018]. This mapping is used to obtain renders of full-body avatars given a multi-view video sequence, which achieves a higher degree of realism than classical systems.

However, training the texture mapping estimator would be impossible without an extensive amount of human-made annotations. In Chapter 4, we show that learning the volumetric nature of the data for the case of human heads is also possible in a purely data-driven way without any annotations and using just the convolutional neural networks. Crucially, our proposed system was capable of creating realistic renders of full human heads, which include the areas of hair and mouth cavity, problematic for the classical rendering methods, in novel views or poses given just a *single image*. In the follow-up work, described in Chapter 5, we also show how a texture mapping, similar to Güler et al. [2018], could be trained in an unsupervised way for the case of human head modeling and subsequently improve the rendering speed of such systems.

While data-driven approaches that use only the 2D convolutional neural networks for image synthesis were shown to have great potential, combining them with volumetric priors could also beneficially affect their robustness and multi-view consistency. Some works Lombardi et al. [2019], Wang et al. [2021] incorporated the weak form of such prior by representing a scene as a sparse or dense volumetric grid. They could then represent a camera motion as a simple transformation of the underlying volume, making a trained model significantly more robust to viewpoint changes. Others have incorporated the geometric constraints on a deeper level by applying neural networks to render and estimate the underlying 3D representations. In

FIGURE 1.2: Selected avatars produced by the methods described in this thesis. Early methods are presented on the left, the outputs generated using more recent megapixel avatars are shown in the center, and mesh-based avatars are on the right. These results are all generated from a single input image.

Chapters 6-7, we describe the representative systems for both of these approaches. First, we describe a system capable of single-shot non-rigid reconstruction of the human heads in the form of a rigged mesh, i.e., a mesh with associated blendshapes and skinning weights. Our method is built upon a face reconstructing parametric model Li et al. [2017] and extends it to handle the hair and shoulder regions in a purely unsupervised way, without ground-truth 3D annotations. Crucially, compared to the previous face reconstruction works, which extend the parametric models Feng et al. [2020], our system is also capable of realistic rendering.

Finally, in Chapter 6, we describe a volumetric rendering system, capable of high-resolution rendering in the challenging *cross-reenactment* scenario. The motion data for the human image synthesis can be either extracted using the images of the same or a different person. Some methods based on deep neural networks Siarohin et al. [2019b], Wang et al. [2021] notably struggle with transferring the motion data obtained from a different person than the one being synthesized. In this work, we propose a principal solution to this problem and develop a method with cross-reenactment performance, substantially improved compared to the competitors. Moreover, we propose a novel way of up-sampling the produced avatars to a high resolution, which achieves less flicker in the video and higher quality than competitors, and also a method of distilling such a system for a fixed number of avatars to achieve real-time performance.

The selected results from the works presented in this thesis are showcased in Figure 1.2.

1.2 Overview

1.2.1 Image Manipulation with Perceptual Discriminators

Image manipulation tasks are usually formalized as a problem of conditional image generation Choi et al. [2018], Isola et al. [2017], Zhu et al. [2017a]. They are formalized as sampling

from the distribution of manipulated realistic images $\hat{x}$, conditioned on the input image x and the variable c that encodes the desired manipulation: $\hat{x} \sim p(\hat{x} \mid x, c)$. The problem of sampling from this distribution can be effectively addressed using a specific architecture of the deep convolutional neural networks, referred to in the literature as image-to-image translation networks Isola et al. [2017]. These deep CNNs are particularly useful since many image manipulation tasks require only local edits at the level of textures.

The choice of training objective is as important for this problem as the usage of image translation networks. The so-called *perceptual loss functions* Johnson et al. [2016b] were shown to be highly efficient at matching the textures and high-level semantics while ignoring some discrepancies, such as misalignment on a pixel level, which are imperceivable to humans. The perceptual loss utilizes intermediate features from a CNN pre-trained for a large-scale image classification task Russakovsky et al. [2015]. Such pre-training allows the network to learn a wide range of useful statistics in its intermediate feature representations. The loss is then defined as the distance between the feature maps, estimated separately for predicted and ground truth images. These losses are used alongside the GAN-based objectives that ensure general realism of the predictions and are shown to be highly effective when combined with image-to-image translation networks Choi et al. [2018], Isola et al. [2017], Zhu et al. [2017a].

However, for some problems, such as facial attribute editing, there are no medium- or large-scale datasets with annotated ground truths for the desired set of manipulations. To be specific, existing datasets consist of pairs of realistic images x and their attributes c, not pairs of images x and $\tilde{x}$ with an imposed manipulation $\tilde{c}$. In this work, we decided to adapt perceptual losses, which are most useful in a supervised training scenario where the ground truth is known, to this unsupervised training setting. Specifically, we propose combining perceptual losses and unsupervised GAN-based losses in a single system. Instead of using raw images as inputs to the discriminator, we use the feature maps of the pre-trained deep CNNs. We call a combination of such a feature extractor and a regular discriminator as a *perceptual discriminator*. The resulting architecture can be utilized in various image manipulation tasks and was shown to benefit from the robustness of perceptual losses.

As stated previously, our proposed system is most effective in the unsupervised setting, specifically an unaligned image-to-image translation problem Zhu et al. [2017a]. In the aligned image-to-image translation, the resulting transformation of the image is known and can be supervised via the abovementioned perceptual losses, reducing the effect of the proposed discriminator. However, in an unaligned translation task, the result of the transform is unknown, and the only learning signal comes from the GAN-based loss function. We have substantially outperformed most existing systems Radford et al. [2015], Zhu et al. [2017a] in this task by simply substituting their discriminator with ours, which was confirmed by an evaluation conducted on multiple datasets.

Such a perceptual discriminator represents a plug-and-play solution for the domains where the usage of perceptual losses is effective. Its main failure case, however, happens when the manipulated images are substantially different from the dataset which is used to train the statistics extractor. For example, while the proposed systems work great for the domain of natural images, the results for their stylization into paintings are on-par with the base methods.

1.2.2 Textured Neural Avatars

The abovementioned image manipulation approach can be quite fruitful for animating human subjects. For example, a number of systems Chan et al. [2018], Wang et al. [2018c] used image-to-image translation techniques to animate a human subject given a dataset of videos depicting its motion and appearance. As an input, such systems use a set of keypoints, which denote subject's pose, and directly output a rendered image. The dataset creation process required to learn such an avatar system can be as simple as capturing a monocular video of a person and processing it via an off-the-shelf keypoints extractor Cao et al. [2017].

However, the failure case of such a naive approach is in its data hungriness. The amount of data required to synthesize various avatar motions in different camera poses grows exponentially since the CNNs are used to produce the 2D images directly. Even a simple linear 3D transformation of the scene, such as rotation, may result in a substantially non-linear 2D transformation of a projected geometry. All combinations of these transformations must be learned by observing the training data, which results in a limited generalization capability of a trained model. While many of these disadvantages could be overcome with the classical mesh-based rendering systems Alldieck et al. [2018a], the resulting renders typically lack the realism produced by the CNN-based systems.

In this work, we present a system that combines the two approaches in a novel formulation. We propose to use a texture-based image synthesis, similar to the classical computer graphics systems, alongside the direct prediction of the rendered texture coordinates similar to the image-to-image translation methods. The estimation of such texture coordinates was pioneered in a DensePose system Güler et al. [2018], which was trained using extensive amount of human-annotated data with image to template mesh correspondenses. As a result of training with noisy and sparsely annotated data, and contrary to classical graphics systems, these texture coordinates do not necessarily correspond to a rendered 3D model, i.e. they do not have to be view-consistent with the camera rotations and translations. Such flexibility allows our system to learn much more realistic renders without tackling the challenging task of reconstructing detailed geometry of the human subject. In our method, we do not rely on the CNNs to synthesize the images directly, and instead predict only the texture coordinates, which greatly increases the generalization of the

resulting system, compared to the approaches that directly synthesize the output images Chan et al. [2018], Wang et al. [2018c].

We first show that our system outperforms the competitors using multi-view training data and then utilize fine-tuning to obtain similar degrees of generalization on the sparse-view or even monocular videos. The main limitation of our method is reliance on pre-trained pose estimation systems Cao et al. [2017] for inference and texture coordinates estimators Güler et al. [2018] for pre-trainining. All errors of these systems are directly translated into our method. Additionally, since CNNs that we use to predict texture coordinates are known to lack scale invariance, the resulting system does not generalize well for the cases when the person is rendered at scales different from the training set. Lastly, the approach that we take does not propperly handles lighting as it lacks view-directional conditioning.

1.2.3 Few-Shot Adversarial Learning of Realistic Neural Talking Head Models

While standard image-to-image translation methods were able to address the problem of avatar creation for individual subjects, such approaches still require a large amount of training data and a lengthy training process. In the previous work, we have shown how to train such systems using sparse-view videos or lengthy monocular videos instead of multi-view data. However, these data and computational requirements are still inadequate for most applications. A more realistic use case would require a short video and minutes of fine-tuning. However, such constraints would require the system to have an extensive human appearance and motion prior. In our follow-up work, we attempted to learn such prior directly from the data using a large-scale dataset of human talking head videos Chung et al. [2018b].

We decided to approach the problem of talking head synthesis using a setting of conditional image generation, similar to our image manipulation work. We treat the problem of avatar generation as sampling from a distribution $p(x \mid y, e)$, where x denotes a rendered image of the avatar for the person e in the pose y encoded in the form of keypoints. The sampler has a form of the neural network G that, conditioned on e, maps y into a sample x. Unlike the previous related image manipulation works, which only tackled a small number of possible conditions Huang et al. [2018], Wang et al. [2018c,g], our method was capable of handling an arbitrary person as an input. Also, compared with the extisting systems of few-shot talking head generation Wiles et al. [2018], our proposed system achieved substantially greater realism.

Inspired by the previous works on style transfer, we utilized adaptive instance normalization layers (AdaIN) Huang and Belongie [2017] to condition our generator with an arbitrary style code. Therefore, all generator parameters can be separated into person-agnostic θ and person-specific $\gamma(e)$. In order to predict person-specific parameters, we use a single image or multiple images depicting a person we want to reenact and pass through an encoding network E. In the case

of multiple images, we obtain the person-specific embeddings for each image individually and then average the results. The predicted embeddings are then used to estimate person-specific parameters $\gamma(e)$ with a multi-layer perceptron (MLP). Such an approach allows us to leverage end-to-end GAN-based training for the problem of avatar creation since all the elements of the pipeline are differentiable.

While this approach achieves a high degree of generalization to novel views and poses, the gap between the person's appearance in the source images and the reenacted images remains quite noticeable. To combat that, we propose an additional fine-tuning step that trains the generator in an autoencoding fashion, i.e., to reenact the images which were used to initialize its person-specific parameters. This approach is reminiscent of model-agnostic meta-learning (MAML) Finn et al. [2017]. The major difference of our work is that we do not modify the training procedure to facilitate the post-training fine-tuning, since that would incur large computational costs. Surprisingly, despite the fact that after fine-tuning a large network with millions of parameters to reenact just a handful of images (as little as one image), it retains its generalization capabilities while improving the personalization of the outputs. We achieve that by keeping the fine-tuning process short to prevent overfitting and by extensively pre-training the network to reenact the avatars without such fine-tuning, as was described previously.

The main limitations of the method include a lack of personalization in the cross-reenactment scenario, where the pose and appearance are estimated using videos of different people. The head pose, being encoded as the keypoints, contains information about the shape of the facial features, which in turn gets imprinted onto the system's output. Another limitation is the inability to handle high-frequency details such as moles and wrinkles, which stems from the lack of capacity of the model.

1.2.4 Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars

While our previous work substantially bridged the gap between the multi-shot avatars and the practical applications, some improvements were still needed to deploy the avatars on edge devices such as smartphones. While improving the avatars' personalization, the fine-tuning process was still relatively slow and unsuitable for deployment on mobile devices since the existing frameworks lack back-propagation capabilities SNP. The inference time per frame could also be improved to achieve real-time performance. Lastly, to make the use-case simpler and broader, we decided to focus on single-shot avatar creation, which was also done in our main competing work Siarohin et al. [2019b].

Our approach models a person's appearance by decomposing it into two layers. The first layer is a pose-dependent coarse image synthesized by a small neural network. The second layer is defined by a pose-independent texture image that contains high-frequency details. The texture

image is generated once per avatar, warped, and added to the coarse image to ensure a higher effective resolution of synthesized head views. Such decomposition allows us to directly produce an RGB texture, similar to the previous works Wiles et al. [2018]. Our main novelty was a neural-based generation of such texture, which allowed the inclusion of high-frequency details and substantially outperformed the previous work in terms of quality. Another novelty, the image decomposition, made the generated results more realistic in handling specular highlights. During initial experiments, we noticed that direct prediction of low-frequency RGB images allowed such highlights to stay in place during head rotation. Lastly, we propose a neural network-based fine-tuning process that achieves a much faster convergence and generalization than stochastic gradient descent. All these improvements combined allow us to handle even small person-specific high-frequency details such as moles, which shows a clear advantage over our previous system in terms of the quality of the outputs.

The main advantage of our system was its real-time capabilities evaluated on Android-based mobile devices, as well as the quality of the results, which outperforms the state-of-the-art methods given the tight computational budget required by real-time performance.

1.2.5 MegaPortraits: One-shot Megapixel Neural Head Avatars

In this work, we advance neural head avatars 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.

To achieve that, 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 show that suggested architectures and methods produce convincing high-resolution neural avatars, outperforming the competitors Doukas et al. [2021b], Siarohin et al. [2019b], Wang et al. [2021] in the cross-driving scenario. To achieve this level of quality, we use the Face Vid-to-vid method Wang et al. [2021] as a base for image synthesis but replace its motion estimation and reenactment components. Instead, we employ self-supervised learning of the motion latent codes, which have a high degree of disentanglement from the latent appearance representations. The idea of using latent motion models was first proposed in the work by Burkov et al. Burkov et al. [2020], while we expand on that idea by proposing a viable method of combining it with larger models by using multiple novel techniques. Lastly, we propose a new way to train a single-image super-resolution module to incorporate high-resolution image-based datasets into the training and boost the quality of the results even further. We show that our semi-supervised training formulation achieves better effective resolution and fewer artifacts in the outputs when applied for out-of-domain cross-reenactment.

We also 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 of head avatar systems.

1.2.6 Realistic one-shot mesh-based avatars

Finally, we tackle the problem of synthesizing personalized avatars using geometry-based rendering. Contrary to our previous works, which mostly ignored the step of estimating the geometry of the subject in favor of directly predicting the images, in this work he ground our rendering with the 3D reconstruction of the human head. Such an approach could be useful for the tasks such as full head tracking, where the full reconstruction of the human subject is required, not just of the facial region, present in the previous works Feng et al. [2020].

The main challenge here is twofold: similarly to our previous works, we want to estimate head avatars using a single image and be able to animate them. This is a highly ill-posed problem: we want to obtain a 3D representation that is regularized enough to estimate it using just a single image and, simultaneously, to have additional degrees of freedom which allow its animation. Previous works Feng et al. [2020], Guo et al. [2020] relied on the existing datasets of 3D scans of human faces, which were distilled into low-dimensional parametric models Blanz and Vetter [1999], Li et al. [2017]. However, none of such datasets were publicly released for full head reconstruction. Moreover, even private data collection effors Wuu et al. [2022] contain only a few hundred subjects, which is insufficient for single-shot head reconstruction due to its diversity of appearance. Therefore, we designed our approach around the learning-by-synthesis paradigm, using a large-scale dataset of videos without any 3D annotation.

Since the existing state-of-the-art face reconstruction methods Feng et al. [2020] rely on strong 3D-based priors, we decided to focus on augmenting their capabilities and reconstructing everything that is missing from them, i.e., the rest of the head, hair, and as a by-product, shoulders. The facial and head rigging would therefore be inherited from these systems, substantially simplifying our problem.

To do that, we propose combining soft rasterization Liu et al. [2019] and deferred neural rendering Thies et al. [2019a] techniques in a novel system designed to work with large-scale training, which in the end, allows learning the shape of the subject from a single observation. This allows us to obtain a 3D reconstruction and a realistic neural-based renderer that we found to outperform previous works Siarohin et al. [2019b], Zakharov et al. [2020].

However, the main limitation of our method is the lack of details in the obtained reconstructions. We manage to only reconstruct a general shape that hair and shoulders occupy, without any details related to clothing or specific hairstyles. Addressing these limitations remains future work.

Chapter 2

Image Manipulation with Perceptual Discriminators

Abstract

Systems that perform image manipulation using deep convolutional networks have achieved remarkable realism. Perceptual losses and losses based on adversarial discriminators are the two main classes of learning objectives behind these advances. In this work, we show how these two ideas can be combined in a principled and non-additive manner. This is accomplished through a special architecture of the discriminator network inside generative adversarial learning framework. The new architecture, that we call a perceptual discriminator, embeds the convolutional parts of a pre-trained deep classification network inside the discriminator network. The resulting architecture can be trained on unaligned image datasets, while benefiting from the robustness and efficiency of perceptual losses. We demonstrate the merits of the new architecture in a series of qualitative and quantitative comparisons with baseline approaches and state-of-the-art frameworks for image manipulations.

This work was published as: Diana Sungatullina*, Egor Zakharov*, Dmitry Ulyanov and Victor Lempitsky. *Image Manipulation with Perceptual Discriminators*. European Conference on Computer Vision (ECCV), 2018. * denotes equal contribution.

Supplementary materials are hosted on the project page: https://egorzakharov.github.io/perceptual_gan

2.1 Introduction

Generative convolutional neural networks have achieved remarkable success in image manipulation tasks both due to their ability to train on large amount of data Dosovitskiy et al. [2015], Jain and Seung [2009], Kim et al. [2016] and due to natural image priors associated with such architectures Ulyanov et al. [2018]. Recently, the ability to train image manipulation ConvNets has been shown in the *unaligned* training scenario Benaim and Wolf [2017], Zhu et al. [2017a,d], where the training is based on sets of images annotated with the presence/absence of a certain attribute, rather than based on *aligned* datasets containing {input,output} image pairs. The ability to train from unaligned data provides considerable flexibility in dataset collection and in learning new manipulation effects, yet poses additional algorithmic challenges.

Generally, the realism of the deep image manipulation methods is known to depend strongly on the choice of the loss functions that are used to train generative ConvNets. In particular, simplistic pixelwise losses (e.g. the squared distance loss) are known to limit the realism and are also non-trivial to apply in the unaligned training scenario. The rapid improvement of realism of deep image generation and processing is thus associated with two classes of loss functions that go beyond pixel-wise losses. The first group (so-called *perceptual losses*) is based on matching activations inside pre-trained deep convolutional networks (the VGG architecture trained for ILSVRC image classification is by far the most popular choice Simonyan and Zisserman [2014]). The second group consists of *adversarial losses*, where the loss function is defined implicitly using a separate *discriminator* network that is trained adversarially in parallel with the main generative network.

The two groups (perceptual losses and adversarial losses) are known to have largely complementary strengths and weaknesses. Thus, perceptual losses are easy to incorporate and are easy to scale to high-resolution images; however, their use in unaligned training scenario is difficult, as these loss terms require a concrete target image to match the activations to. Adversarial losses have the potential to achieve higher realism and can be used naturally in the unaligned scenarios, yet adversarial training is known to be hard to set up properly, often suffers from mode collapse, and is hard to scale to high-resolution images. Combining perceptual and adversarial losses in an additive way has been popular Dosovitskiy and Brox [2016], Ledig et al. [2017b], Sajjadi et al. [2017], Wang et al. [2017]. Thus, a generative ConvNet can be trained by minimizing a linear combination of an adversarial and a perceptual (and potentially some other) losses. Yet such additive combination includes not only strengths but also weaknesses of the two approaches. In particular, the use of a perceptual loss still incurs the use of aligned datasets for training.

In this work we present an architecture for realistic image manipulation, which combines perceptual and adversarial losses in a natural *non-additive* way. Importantly, the architecture keeps the ability of adversarial losses to train on unaligned datasets, while also benefits from the stability

of perceptual losses. Our idea is very simple and concerned with the particular design of the discriminator network for adversarial training. The design encapsulates a pretrained classification network as the initial part of the discriminator. During adversarial training, the generator network is effectively learned to match the activations inside several layers of this reference network, just like the perceptual losses do. We show that the incorporation of the pretrained network into the discriminator stabilizes the training and scales well to higher resolution images, as is common with perceptual losses. At the same time, the use of adversarial training allows to avoid the need for aligned training data.

Generally, we have found that the suggested architecture can be trained with little tuning to impose complex image manipulations, such as adding to and removing smile from human faces, face ageing and rejuvenation, gender change, hair style change, etc. In the experiments, we show that our architecture can be used to perform complex manipulations at medium and high resolutions, and compare the proposed architecture with several adversarial learning-based baselines and recent methods for learning-based image manipulation.

2.2 Related work

2.2.1 Generative ConvNets.

Our approach is related to a rapidly growing body of works on ConvNets for image generation and editing. Some of the earlier important papers on ConvNet image generation Dosovitskiy et al. [2015] and image processing Dong et al. [2014], Jain and Seung [2009], Kim et al. [2016] used per-pixel loss functions and fully supervised setting, so that at test time the target image is known for each input. While this demonstrated the capability of ConvNets to generate realistic images, the proposed systems all had to be trained on aligned datasets and the amount of high-frequency details in the output images was limited due to deficiencies of pixel-wise loss functions.

2.2.2 Perceptual Losses.

The work of Mahendran and Vedaldi Mahendran and Vedaldi [2015] has demonstrated that the activations invoked by an image within a pre-trained convolutional network can be used to recover the original image. Gatys et al. Gatys et al. [2016] showed that such activations can serve as content descriptors or texture descriptors of the input image, while Dosovitsky and Brox Dosovitskiy and Brox [2016], Ulyanov et al. Ulyanov et al. [2016], Johnson et al. Johnson et al. [2016a] have shown that the mismatches between the produced and the target activations can be used as so-called *perceptual losses* for a generative ConvNet. The recent work of Chen and Koltun [2017] pushed the spatial resolution and the realism of images produced by a feed-forward

ConvNet with perceptual losses to megapixel resolution. Generally, in all the above-mentioned works Chen and Koltun [2017], Dosovitskiy and Brox [2016], Johnson et al. [2016a], Ulyanov et al. [2016], the perceptual loss is applied in a fully supervised manner as for each training example the specific target deep activations (or the Gram matrix thereof) are given explicitly. Finally, Upchurch et al. [2017] proposed a method that manipulates carefully aligned face images at high resolution by compositing desired activations of a deep pretrained network and finding an image that matches such activations using the non-feedforward optimization process similar to Gatys et al. [2016], Mahendran and Vedaldi [2015].

2.2.3 Adversarial Training.

The most impressive results of generative ConvNets were obtained within generative adversarial networks (GANs) framework proposed originally by Goodfellow et al. Goodfellow et al. [2014a]. The idea of adversarial training is to implement the loss function as a separate trainable network (the *discriminator*), which is trained in parallel and in adversarial way with the generative ConvNet (the *generator*). Multiple follow-up works including Arjovsky et al. [2017], Karras et al. [2017], Radford et al. [2015], Salimans et al. [2016] investigated the choice of convolutional architectures for the generator and for the discriminator. Achieving reliable and robust convergence of generator-discriminator pairs remains challenging Chintala et al. [2017], Goodfellow [2017], Lucic et al. [2017], and in particular requires considerably more efforts than training with perceptual loss functions.

2.2.4 Unaligned Adversarial Training.

While a lot of the original interest to GANs was associated with unconditional image generation, recently the emphasis has shifted to the conditional image synthesis. Most relevant to our work are adversarially-trained networks that perform image translation, i.e. generate output images conditioned on input images. While initial methods used aligned datasets for training Isola et al. [2017], Zhang et al. [2016a], recently some impressive results have been obtained using unaligned training data, where only empirical distributions of the input and the output images are provided Benaim and Wolf [2017], Zhu et al. [2017a,d]. For face image manipulation, systems using adversarial training on unaligned data have been proposed in Brock et al. [2016], Choi et al. [2018]. While we also make an emphasis on face manipulation, our contribution is orthogonal to Brock et al. [2016], Choi et al. [2018] as perceptual discriminators can be introduced into their systems.

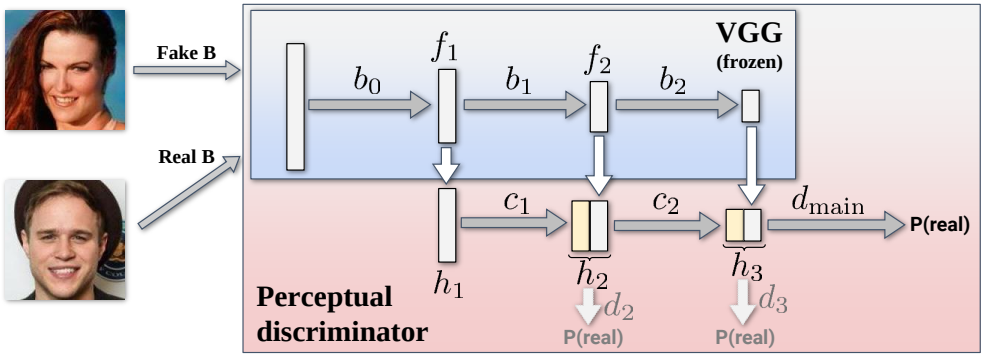


FIGURE 2.1: The perceptual discriminator is composed of a pre-trained image classification network (such as VGG), split into blocks b_i. The parameters of those blocks are not changed during training, thus the discriminator retains access to so-called perceptual features. The outputs of these blocks are processed using learnable blocks of convolutional operations c_i and the outputs of those are used to predict the probability of an image being real or manipulated (the simpler version uses a single discriminator d_{main}, while additional path discriminators are used in the full version).

2.2.5 Combining Perceptual and Adversarial Losses.

A growing number of works Dosovitskiy and Brox [2016], Ledig et al. [2017b], Wang et al. [2017] use the combination of perceptual and adversarial loss functions to accomplish more stable training and to achieve convincing image manipulation at high resolution. Most recently, Sajjadi et al. [2017] showed that augmenting perceptual loss with the adversarial loss improves over the baseline system Chen and Koltun [2017] (that has already achieved very impressive results) in the task of megapixel-sized conditional image synthesis. Invariably, the combination of perceptual and adversarial losses is performed in an additive manner, i.e. the two loss functions are weighted and added to each other (and potentially to some other terms). While such additive combination is simple and often very efficient, it limits learning to the aligned scenario, as perceptual terms still require to specify target activations for each training example. In this work, we propose a natural non-additive combination of perceptual losses and adversarial training that avoids the need for aligned data during training.

2.3 Method

2.3.1 Background and motivation

Generative adversarial networks have shown impressive results in photorealistic image synthesis. The model includes a generative network G, that is trained to match the target distribution $p_{\text{target}}(\mathbf{y})$ in the data space $\mathcal{Y}$, and a discriminator network D that is trained to distinguish whether the input is real or generated by G. In the simplest form, the two networks optimize the policy

function $V(D, G)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{y} \sim p_{\text{target}}(\mathbf{y})} \log D(\mathbf{y}) + \mathbb{E}_{\mathbf{x} \sim p_{\text{source}}(\mathbf{x})} [\log(1 - D(G(\mathbf{x}))], \tag{2.1}$$

In (2.1), the source distribution $p_{\text{source}}(\mathbf{x})$ may correspond to a simple parametric distribution in a latent space such as the unit Gaussian, so that after training unconditional samples from the learned approximation to $p_{\text{target}}(\mathbf{y})$ can be drawn. Alternatively, $p_{\text{source}}(\mathbf{x})$ may correspond to another empirical distribution in the image space $\mathcal{X}$. In this case, the generator learns to *translate* images from $\mathcal{X}$ to $\mathcal{Y}$, or to *manipulate* images in the space $\mathcal{X}$ (when it coincides with $\mathcal{Y}$). Although our contribution (perceptual discriminators) is applicable to both unconditional synthesis and image manipulation/translation, we focus our evaluation on the latter scenario. For the low resolution datasets, we use the standard non-saturating GAN modification, where the generator maximizes the log-likelihood of the discriminator instead of minimizing the objective (2.1) Goodfellow et al. [2014a]. For high-resolution images, following CycleGAN Zhu et al. [2017a], we use the LSGAN formulation Mao et al. [2016].

Converging to good equilibria for any of the proposed GAN games is known to be hard Chintala et al. [2017], Goodfellow [2017], Lucic et al. [2017]. In general, the performance of the trained generator network crucially depends on the architecture of the discriminator network, that needs to learn meaningful statistics, which are good for matching the target distribution p_{target}. The typical failure mode of GAN training is when the discriminator does not manage to learn such statistics before being "overpowered" by the generator.

2.3.2 Perceptual Discriminator Architecture

Multiple approaches have suggested to use activations invoked by an image $\mathbf{y}$ inside a deep pre-trained classification network $F(\mathbf{y})$ as statistics for such tasks as retrieval Babenko et al. [2014] or few-shot classification Razavian et al. [2014]. Mahendran and Vedaldi Mahendran and Vedaldi [2015] have shown that activations computed after the convolutional part of such network retain most of the information about the input $\mathbf{y}$, i.e. are essentially invertable. Subsequent works such as Dosovitskiy and Brox [2016], Gatys et al. [2016], Johnson et al. [2016a], Ulyanov et al. [2016] all used such "perceptual" statistics to match low-level details such as texture content, certain image resolution, or particular artistic style.

Following this line of work, we suggest to base the GAN discriminator $D(\mathbf{y})$ on the perceptual statistics computed by the reference network F on the input image $\mathbf{y}$, which can be either real (coming from p_{target}) or fake (produced by the generator). Our motivation is that a discriminator that uses perceptual features has a better chance to learn good statistics than a discriminator initialized to a random network. For simplicity, we assume that the network F has a chain structure, e.g. F can be the VGGNet of Simonyan and Zisserman [2014].

Consider the subsequent blocks of the convolutional part of the reference network F, and denote them as $b_0, b_1, \ldots, b_{K-1}$. Each block may include one or more convolutional layers interleaved with non-linearities and pooling operations. Then, the perceptual statistics $\{f_1(\mathbf{y}), \ldots, f_K(\mathbf{y})\}$ are computed as:

$$f_1(\mathbf{y}) = b_0(\mathbf{y}) \tag{2.2}$$

$$f_i(\mathbf{y}) = b_{i-1}(f_{i-1}(\mathbf{y})), \quad i = 2, \ldots, K\,, \tag{2.3}$$

so that each $f_i(\mathbf{y})$ is a stack of convolutional maps of the spatial dimensions $W_i \times W_i$. The dimension W_i is determined by the preceeding size W_{i-1} as well as by the presence of strides and pooling operations inside b_i. In our experiments we use features from consecutive blocks, i.e. $W_i = W_{i-1}/2$.

The overall structure of our discriminator is shown in Figure 2.1. The key novelty of our discriminator is the in-built perceptual statistics f_i (top of the image), which are known to be good at assessing image realism Gatys et al. [2016], Johnson et al. [2016a], Upchurch et al. [2017]. During the backpropagation, the gradients to the generator flow through the perceptual statistics extractors b_i, but the parameters of b_i are frozen and inherited from the network pretrained for large-scale classification. This stabilizes the training, and ensures that at each moment of time the discriminator has access to "good" features, and therefore cannot be overpowered by the generator easily.

In more detail, the proposed discriminator architecture combines together perceptual statistics using the following computations:

$$h_1(\mathbf{y}) = f_1(\mathbf{y}) \tag{2.4}$$

$$h_i(\mathbf{y}) = \texttt{stack}\left[c_{i-1}(h_{i-1}(\mathbf{y}), \phi_{i-1})\,,\ f_i(\mathbf{y})\right], \quad i = 2, \ldots, K\,, \tag{2.5}$$

where `stack` denotes stacking operation, and the convolutional blocks c_j with learnable parameters ϕ_j (for $j = 1, \ldots, K-1$) are composed of convolutions, leaky ReLU nonlinearities, and average pooling operations. Each of the c_j blocks thus transforms map stacks of the spatial size $W_j \times W_j$ to map stacks of the spatial size $W_{j+1} \times W_{j+1}$. Thus, the strides and pooling operations inside c_j match the strides and/or pooling operations inside b_j.

Using a series of convolutional and fully-connected layers with learnable parameters ψ_{main} applied to the representation $h_K(\mathbf{y})$, the discriminator outputs the probability d_{main} of the whole image $\mathbf{y}$ being real. For low- to medium- resolution images we perform experiments using only this probability. For high-resolution, we found that additional outputs from the discriminator resulted in better outcomes. Using the "patch discriminator" idea Isola et al. [2017], Zhu et al. [2017a], to several feature representations h_j we apply a convolution+LeakyReLU block d_j with learnable parameters ψ_j that outputs probabilities $d_{j,p}$ at every spatial locations p. We then

replace the regular log probability $\log D(\mathbf{y}) \equiv \log d_{\text{main}}$ of an image being real with:

$$\log D(\mathbf{y}) = \log d_{\text{main}}(\mathbf{y}) + \sum_j \sum_{p \in \text{Grid}(W_j \times W_j)} \log d_{j,p}(\mathbf{y}) \tag{2.6}$$

Note, that this makes our discriminator "multi-scale", since spatial resolution W_j varies for different j. The idea of multiple classifiers inside the discriminator have also been proposed recently in Iizuka et al. [2017], Wang et al. [2017]. Unlike Iizuka et al. [2017], Wang et al. [2017] where these classifiers are disjoint, in our architecture all such classifiers are different branches of the same network that has perceptual features underneath.

During training, the parameters of the c blocks inside the feature network F remain fixed, while the parameters ϕ_i of feature extractors c_i and the parameters ψ_i of the discriminators d_i are updated during the adversarial learning, which forces the "perceptual" alignment between the output of the generator and p_{target}. Thus, wrapping perceptual loss terms into additional layers c_i and d_i and putting them together into the adversarial discriminator allows us to use such perceptual terms in the unaligned training scenario. Such unaligned training was, in general, not possible with the "traditional" perceptual losses.

2.3.3 Architecture Details

Reference Network. Following multiple previous works Gatys et al. [2016], Johnson et al. [2016a], Ulyanov et al. [2016], we consider the so-called *VGG network* from Simonyan and Zisserman [2014] trained on ILSVRC2012 Russakovsky et al. [2014] as the reference network F. In particular, we pick the VGG-19 variant, to which we simply refer to as VGG. While the perceptual features from VGG already work well, the original VGG architecture can be further improved. Radford et. al Radford et al. [2015] reported that as far as leaky ReLU avoids sparse gradients, replacing ReLUs with leaky ReLUs He et al. [2015b] in the discriminator stabilizes the training process of GANs. For the same reasons, changing max pooling layers to average pooling removes unwanted sparseness in the backpropagated gradients. Following these observations, we construct the *VGG** network, which is particularly suitable for the adversarial game. We thus took the VGG-19 network pretrained on ILSVRC dataset, replaced all max pooling layers by average poolings, ReLU nonlinearities by leaky ReLUs with a negative slope 0.2 and then trained on the ILSVRC dataset for further two days. We compare the variants of our approach based on VGG and VGG* features below.

Generator Architecture. For the image manipulation experiments, we used transformer network proposed by Johnson et al. Johnson et al. [2016a]. It consists of M convolutional layers with stride size 2, N residual blocks He et al. [2015a] and M upsampling layers, each

one increases resolution by a factor of 2. We set M and N in a way that allows outputs of the last residual block to have large enough receptive field, but at the same time for generator and discriminator to have similar number of parameters. We provide detailed descriptions of architectures in Sungatullina et al. [2018b].

Stabilizing the Generator. We have also used two additional methods to improve the generator learning and to prevent its collapse. First, we have added the *identity loss* Taigman et al. [2016], Zhu et al. [2017a] that ensures that the generator does not change the input, when it comes from the p_{target}. Thus, the following term is added to the maximization objective of the generator:

$$J_{\text{id}}^{G} = -\lambda_{\text{id}}\, \mathbb{E}_{\mathbf{y}\sim p_{\text{target}}} \lambda \left\| \mathbf{y} - G(\mathbf{y}) \right\|_{L_1}, \tag{2.7}$$

where λ_{id} is a meta-parameter that controls the contribution of the weight, and $\|\cdot\|_{L_1}$ denotes pixel-wise L1-metric.

To achieve the best results for the hardest translation tasks, we have found the cycle idea from the CycleGAN Zhu et al. [2017a] needed. We thus train two generators $G_{\mathbf{x}\to\mathbf{y}}$ and $G_{\mathbf{y}\to\mathbf{x}}$ operating in opposite directions in parallel (and jointly with two discriminators), while adding reciprocity terms ensuring that mappings $G_{\mathbf{x}\to\mathbf{y}} \circ G_{\mathbf{y}\to\mathbf{x}}$ and $G_{\mathbf{y}\to\mathbf{x}} \circ G_{\mathbf{x}\to\mathbf{y}}$ are close to identity mappings.

Moreover, we notice that usage of external features as inputs for the discriminator leads to fast convergence of the discriminator loss to zero. Even though this is expected, since our method essentially corresponds to pretraining of the discriminator, this behavior is one of the GAN failure cases Chintala et al. [2017] and on practice leads to bad results in harder tasks. Therefore we find pretraining of the generator to be required for increased stability. For image translation task we pretrain generator as autoencoder. Moreover, the necessity to pretrain the generator makes our approach fail to operate in DCGAN setting with unconditional generator.

After an additional stabilization through the pretraining and the identity and/or cycle losses, the generator becomes less prone to collapse. Overall, in the resulting approach it is neither easy for the discriminator to overpower the generator (this is prevented by the identity and/or cycle losses), nor is it easy for the generator to overpower the discriminator (as the latter always has access to perceptual features, which are good at judging the realism of the output).

2.4 Experiments

The goal of the experimental validation is two-fold. The primary goal is to validate the effect of perceptual discriminators as compared to baseline architectures which use traditional discriminators that do not have access to perceptual features. The secondary goal is to validate the

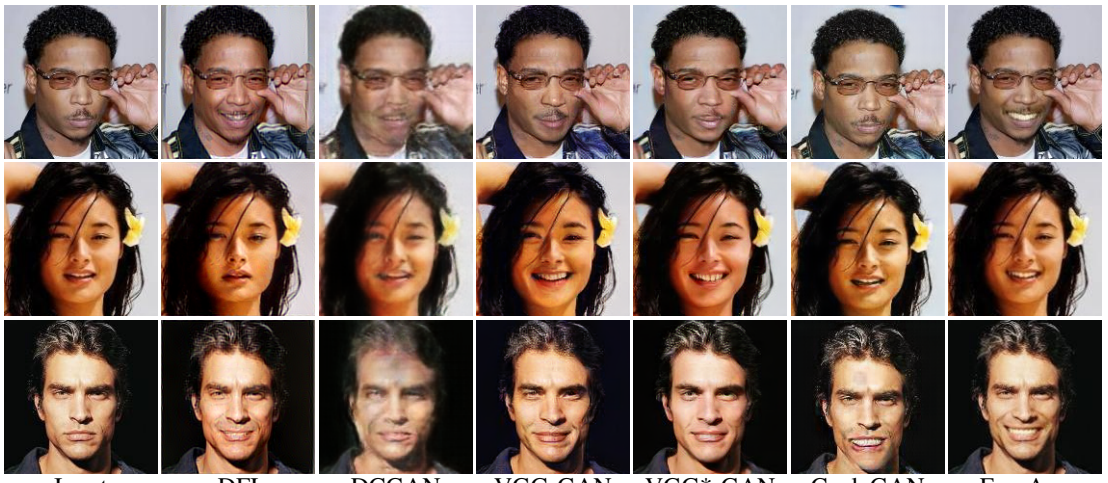


FIGURE 2.2: Qualitative comparison of the proposed systems as well as baselines for neutral→smile image manipulation. As baselines, we show the results of DFI (perceptual features, no adversarial training) and DCGAN (same generator, no perceptual features in the discriminator). Systems with perceptual discriminators output more plausible manipulations.

ability of our full system based on perceptual discriminators to handle harder image translation/-manipulation task with higher resolution and with less data. Extensive additional results are available on our project page Sungatullina et al. [2018b]. We perform the bulk of our experiments on CelebA dataset Liu et al. [2015], due to its large size, popularity and the availability of the attribute annotations (the dataset comprises over 200k of roughly-aligned images with 40 binary attributes; we use 160×160 central crops of the images). As harder image translation task, we use CelebA-HQ Karras et al. [2017] dataset, which consists of high resolution versions of images from CelebA and is smaller in size. Lastly, we evaluate our model on problems with non-face datasets like apples to oranges and photo to Monet texture transfer tasks.

Experiments were carried out on NVIDIA DGX-2 server.

2.4.1 Qualitative Comparison on CelebA.

Even though our contribution is orthogonal to a particular GAN-based image translation method, we chose one of them, provided modifications we proposed and compared it with the following important baselines in an attribute manipulation task:

- *DCGAN* Radford et al. [2015]: in this baseline GAN system we used image translation model with generator and discriminator trained only with adversarial loss.
- *CycleGAN* Zhu et al. [2017a]: this GAN-based method learns two reciprocal transforms in parallel with two discriminators in two domains. We have used the authors' code (PyTorch version).

TABLE 2.1: Quantitative comparison: (a) Photorealism user study. We show the fraction of times each method has been chosen as "the best" among all in terms of photorealism and identity preservation (the higher the better). (b) C2ST results (cross-entropy, the higher the better). (c) Log-loss of classifier trained on real data for each class (the lower the better). See main text for details.

	(a) User study		(b) C2ST, $\times 10^{-2}$			(c) Classification loss		
	Smile	Age	Smile	Gender	Hair color	Smile	Gender	Hair color
DFI Upchurch et al. [2017]	0.16	0.4	< 0.1	< 0.01	< 0.01	1.3	0.5	1.14
FaceApp FaceApp [2018]	0.45	0.41	–	–	–	–	–	–
DCGAN Radford et al. [2015]	–	–	0.6	0.03	0.06	0.6	1.5	2.33
CycleGAN Zhu et al. [2017a]	0.03	0.04	5.3	0.35	0.49	1.2	0.8	2.41
VGG-GAN	–	–	8.6	0.21	0.96	0.4	0.1	1.3
VGG*-GAN	0.36	0.15	5.2	0.24	1.29	0.7	0.1	1.24
Real data	–	–	–	–	–	0.1	0.01	0.56

- *DFI* Upchurch et al. [2017]: to transform an image, this approach first determines target VGG feature representation by adding the feature vector corresponding to input image and the shift vector calculated using nearest neighbours in both domains. Then the resulting image is produced using optimization-based feature inversion as in Mahendran and Vedaldi [2015]. We have used the authors' code.

- *FaceApp* FaceApp [2018]: is a very popular closed-source app that is known for the quality of its filters (transforms), although the exact algorithmic details are unknown.

Our model is represented by two basic variants.

- *VGG-GAN*: we use DCGAN as our base model. The discriminator has a single classifier and no generator pretraining or regularization is applied, other than identity loss mentioned in the previous section.

- *VGG*-GAN*: same as the previous model, but we use a finetuned VGG network variant with dense gradients.

The comparison with state-of-the-art image transformation systems is performed to verify the competitiveness of the proposed architecture (Figure 2.2). In general, we observe that VGG*-GAN and VGG-GAN models consistently outperformed DCGAN variant, achieving higher effective resolution and obtaining more plausible high-frequency details in the resulting images. While a more complex CycleGAN system is also capable of generating crisp images, we found that the synthesized smile often does not look plausible and does not match the face. DFI turns out to be successful in attribute manipulation, yet often produces undesirable artifacts, while FaceApp shows photorealistic results, but with low attribute diversity. Here we also evaluate the contribution of dense gradients idea for VGG encoder and find it providing minor quality improvements.

FIGURE 2.3: Results for VGG*-MS-CycleGAN attribute editing at 256 × 256 resolution on Celeba-HQ dataset. Networks have been trained to perform pairwise domain translation between the values of hair color, gender and smile attributes. Digital zoom-in recommended. See Sungatullina et al. [2018b] for more manipulation examples.

2.4.2 User Photorealism Study on CelebA.

We have also performed an informal user study of the photorealism. The study enrolled 30 subjects unrelated to computer vision and evaluated the photorealism of VGG*-GAN, DFI, CycleGAN and FaceApp on smile and aging/rejuvenation transforms. To assess the photorealism, the subjects were presented quintuplets of photographs unseen during training. In each quintuplet the center photo was an image without the target attribute (e. g. real photo of neutral expression), while the other four pictures were manipulated by one of the methods and presented in random order. The subjects were then asked to pick one of the four manipulations that they found most plausible (both in terms of realism and identity preservation). While there was no hard time limit, the users were asked to make the pick as quickly as possible. Each subject was presented overall 30 quintuplets with 15 quantuplets allocated for each of the considered attribute. The results in Table 3.1a show that VGG*-GAN is competitive and in particular considerably better than the other feed-forward method in the comparison (CycleGAN), but FaceApp being the winner overall. This comes with the caveat that the training set of FaceApp is likely to be bigger than CelebA. We also speculate that the diversity of smiles in FaceApp seems to be lower (Figure 2.2), which is the deficiency that is not reflected in this user study.

2.4.3 Quantitative Results on CelebA.

To get objective performance measure, we have used the classifier two-sample test (C2ST) Lopez-Paz and Oquab [2016] to quantitatively compare GANs with the proposed discriminators to other methods. For each method, we have thus learned a separate classifier to discriminate between hold-out set of real images from target distribution and synthesized images, produced by each of the methods. We split both hold-out set and the set of fake images into training and testing parts, fit the classifier to the training set and report the log-loss over the testing set in the Table 2.1b. The results comply with the qualitative observations: artifacts, produced by DCGAN and DFI are being easily detected by the classifier resulting in a very low log-loss. The proposed system stays on par with a more complex CycleGAN (better on two transforms out of three), proving that a perceptual discriminator can remove the need in two additional networks and cycle losses. Additionally, we evaluated attribute translation performance in a similar fashion to StarGAN Choi et al. [2018]. We have trained a model for attribute classification on CelebA and measured average log-likelihood for the synthetic and real data to belong to the target class. Our method achieved lower log-loss than other methods on two out of three face attributes (see Table 2.1c).

2.4.4 Higher Resolution.

We further evaluate our model on CelebA-HQ dataset. Here in order to obtain high quality results we use all proposed regularization methods. We refer to our best model as VGG*-MS-CycleGAN, which corresponds to the usage of VGG* network with dense gradients as an encoder, multi-scale perceptual discriminator based on VGG* network, CycleGAN regularization and pretraining of the generator. Following CycleGAN, we use LSGAN Mao et al. [2016] as an adversarial objective for that model. We trained on 256×256 version of CelebA-HQ dataset and present attribute manipulation results in Figure 2.3. As we can see, our model provides photorealistic samples while capturing differences between the attributes even for smaller amount of training samples (few thousands per domain) and higher resolution compared to our previous tests.

In order to ensure that each of our individual contributions affects the quality of these results, we consider three variations of our discriminator architecture and compare them to the alternative multi-scale discriminator proposed in Wang et al. Wang et al. [2017]. While Wang et al. used multiple identical discriminators operating at different scales, we argue that this architecture has redundancy in terms of number of parameters and can be reduced to our architecture by combining these discriminators into a single network with shared trunk and separate multi-scale output branches (as is done in our method). Both variants are included into the comparison in Figure 2.4. Also we consider *Rand-MS-CycleGAN* baseline that uses random weights in the feature extractor in order to tease apart the contribution of VGG* architecture as a feature network F and the effect of also having its weights pretrained on the success of the adversarial

(a) (b) (c) (d) (e) (a) (b) (c) (d) (e)

FIGURE 2.4: We compare different architectures for the discriminator on CelebA-HQ 256×256 male $\leftrightarrow$ female problem. We train all architectures in CycleGAN manner with LSGAN objective and compare different discriminator architectures. (a) Input, (b) VGG*-MS-CycleGAN: multi-scale perceptual discriminator with pretrained VGG* as a feature network F, (c) Rand-MS-CycleGAN: multi-scale perceptual discriminator with a feature network F having VGG* architecture with randomly-initialized weights, (d) MS-CycleGAN: multi-scale discriminator with the trunk shared across scales (as in our framework), where images serve as a direct input, (e) separate multi-scale discriminators similar to Wang et al. Wang et al. [2017]. Digital zoom-in recommended.

Input CycleGAN VGG*-CycleGAN Input CycleGAN VGG*-CycleGAN

FIGURE 2.5: Comparison between CycleGAN and VGG*-MS-CycleGAN on painting$\leftrightarrow$photo translation task. It demonstrates the applicability of our approach beyond face image manipulation. See Sungatullina et al. [2018b] for more examples.

training. While the weights inside the VGG part were not frozen, so that adversarial training process could theoretically evolve good features in the discriminator, we were unable to make this baseline produce reasonable results. For high weight of the identity loss λ_{id} the resulting generator network produced near-identical results to the inputs, while decreasing λ_{id} lead to severe generator collapse. We conclude that the architecture alone cannot explain the good performance of perceptual discriminators (which is validated below) and that having pretrained weights in the feature network is important.

2.4.5 Non-face Datasets.

While the focus of our evaluation was on face attribute modification tasks, our contribution applies to other translation tasks, as we verify in this section by performing qualitative comparison with

FIGURE 2.6: Apple↔orange translation samples with CycleGAN and VGG*-MS-CycleGAN are shown. Zoom-in recommended. See Sungatullina et al. [2018b] for more examples.

the CycleGAN and VGG*-MS-CycleGAN architectures on two non-face domains on which CycleGAN was originally evaluated: an artistic style transfer task (Monet-photographs) in Figure 2.5 and an apple-orange conversion in Figure 2.6 (the figures show representative results). To achieve fair comparison, we use the same amount of residual blocks and channels in the generator and the same number of downsampling layers and initial amount of channels in discriminator both in our model and in the original CycleGAN. We used the authors' implementation of CycleGAN with default parameters. While the results on the style transfer task are inconclusive, for the harder apple-to-orange task we generally observe the performance of perceptual discriminators to be better.

2.4.6 Other Learning Formulations.

Above, we have provided the evaluation of the perceptual discriminator idea to unaligned image translation tasks. In principle, perceptual discriminators can be used for other tasks, e.g. for unconditional generation and aligned image translation. In our preliminary experiments, we however were not able to achieve improvement over properly tuned baselines. In particular, for aligned image translation (including image superresolution) an additive combination of standard discriminator architectures and perceptual losses performs just as well as our method. This is not surprising, since the presence of alignment means that perceptual losses can be computed straight-forwardly, while they also stabilize the GAN learning in this case. For unconditional image generation, a naive application of our idea leads to discriminators that quickly overpower generators in the initial stages of the game leading to learning collapse.

2.5 Summary

We have presented a new discriminator architecture for adversarial training that incorporates perceptual loss ideas with adversarial training. We have demonstrated its usefulness for unaligned

image translation tasks, where the direct application of perceptual losses is infeasible. Our approach can be regarded as an instance of a more general idea of using transfer learning, so that easier discriminative learning formulations can be used to stabilize and improve GANs and other generative learning formulations.

Chapter 3

Textured Neural Avatars

Abstract

We present a system for learning full-body *neural avatars*, i.e. deep networks that produce full-body renderings of a person for varying body pose and camera position. Our system takes the middle path between the classical graphics pipeline and the recent deep learning approaches that generate images of humans using image-to-image translation. In particular, our system estimates an explicit two-dimensional texture map of the model surface. At the same time, it abstains from explicit shape modeling in 3D. Instead, at test time, the system uses a fully-convolutional network to directly map the configuration of body feature points w.r.t. the camera to the 2D texture coordinates of individual pixels in the image frame. We show that such a system is capable of learning to generate realistic renderings while being trained on videos annotated with 3D poses and foreground masks. We also demonstrate that maintaining an explicit texture representation helps our system to achieve better generalization compared to systems that use direct image-to-image translation.

This work was published as: Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and Victor Lempitsky *Textured Neural Avatars*. Computer Vision and Pattern Recognition (CVPR), 2019.

Supplementary materials are hosted on the project page: https://samsunglabs.github.io/textured_avatar

3.1 Introduction

Capturing and rendering human body in all of its complexity under varying pose and imaging conditions is one of the core problems of both computer vision and computer graphics. Recently, there is a surge of interest that involves deep convolutional networks (ConvNets) as an alternative to traditional computer graphics means. Realistic *neural rendering* of body fragments e.g. faces Kim et al. [2018a], Lombardi et al. [2018a], Suwajanakorn et al. [2017], eyes Ganin et al. [2016], hands Mueller et al. [2018] is now possible. Very recent works have shown the abilities of such networks to generate views of a person with a varying body pose but with a fixed camera position, and using an excessive amount of training data Aberman et al. [2018], Chan et al. [2018], Liu et al. [2018], Wang et al. [2018b]. In this work, we focus on the learning of *neural avatars*, i.e. generative deep networks that are capable of rendering views of individual people under varying body pose defined by a set of 3D positions of the body joints and under varying camera positions (Figure 4.1). We prefer to use body joint positions to represent the human pose, as joint positions are often easier to capture using marker-based or marker-less motion capture systems.

Generally, neural avatars can serve as an alternative to classical ("neural-free") avatars based on a standard computer graphics pipeline that estimates a user-personalized body mesh in a neutral position, performs skinning (deformation of the neutral pose), and projects the resulting 3D surface onto the image coordinates, while superimposing person-specific 2D texture. Neural avatars attempt to shortcut the multiple stages of the classical pipeline and to replace them with a single network that learns the mapping from the input (the location of body joints) to the output (the 2D image). As a part of our contribution, we demonstrate that, however appealing for its conceptual simplicity, existing pose-to-image translation networks generalize poorly to new camera views, and therefore new architectures for neural avatars are required.

Towards this end, we present a neural avatar system that does full-body rendering and combines the ideas from the classical computer graphics, namely the decoupling of geometry and texture, with the use of deep convolutional neural networks. In particular, similarly to the classic pipeline, our system explicitly estimates the 2D textures of body parts. The 2D texture within the classical pipeline effectively transfers the appearance of the body fragments across camera transformations and body articulations. Keeping this component within the neural pipeline boosts generalization across such transforms. The role of the convolutional network in our approach is then confined to predicting the texture coordinates of individual pixels in the output 2D image given the body pose and the camera parameters (Figure 3.1). Additionally, the network predicts the body foreground/background mask.

In our experiments, we compare the performance of our *textured neural avatar* with a direct video-to-video translation approach Wang et al. [2018b], and show that explicit estimation of

textures brings additional generalization capability and improves the realism of the generated images for new views and/or when the amount of training data is limited.

3.2 Related work

Our approach is closely related to a vast number of previous works, and below we discuss a small subset of these connections.

3.2.1 Full-body avatars

Building avatars from image data has long been one of the main subsections of computer vision research. Traditionally, an avatar is defined by a 3D geometric mesh of a certain neutral pose, a texture, and a skinning mechanism that transforms the mesh vertices according to pose changes. A large group of works has been devoted to body modeling from 3D scanners Pons-Moll et al. [2015], registered multi-view sequences Robertini et al. [2017] as well as from depth and RGB-D sequences Bogo et al. [2015], Weiss et al. [2011], Yu et al. [2018]. On the other extreme are methods that fit skinned parametric body models to single images Bălan and Black [2008], Bogo et al. [2016], Hasler et al. [2010], Kanazawa et al. [2018], Omran et al. [2018], Pavlakos et al. [2018], Starck and Hilton [2003]. Finally, research on building full-body avatars from monocular videos has started Alldieck et al. [2018b,c]. Similarly to the last group of works, our work builds an avatar from a video or a set of unregistered monocular videos. The classical (computer graphics) approach to modeling human avatars requires explicit physically-plausible modeling of human skin, hair, sclera, clothing surface, as well as motion under pose changes. Despite considerable progress in reflectivity modeling Alexander et al. [2010b], Donner et al. [2008], Klehm et al. [2015], Weyrich et al. [2006], Wood et al. [2015] and better skinning/dynamic surface modeling Feng et al. [2015], Loper et al. [2015], Stavness et al. [2014], the computer graphics approach still requires considerable "manual" effort of designers to achieve high realism Alexander et al. [2010b] and to pass the so-called uncanny valley Mori [1970], especially if real-time rendering of avatars is required.

3.2.2 Image synthesis using deep convolutional neural networks

A lot of recent efforts Dosovitskiy et al. [2015], Goodfellow et al. [2014a] has been directed onto synthesis of realistic human faces Choi et al. [2018], Karras et al. [2018], Sungatullina et al. [2018a]. Compared to traditional computer graphics representations, deep ConvNets model data by fitting an excessive number of learnable weights to training data. Such ConvNets avoid explicit modeling of the surface geometry, surface reflectivity, or surface motion under pose changes, and

therefore do not suffer from the lack of realism of the corresponding components. On the flipside, the lack of ingrained geometric or photometric models in this approach means that generalizing to new poses and in particular to new camera views may be problematic. Still a lot of progress has been made over the last several years for the neural modeling of personalized talking head models Kim et al. [2018a], Lombardi et al. [2018a], Suwajanakorn et al. [2017], hair Wei et al. [2018], hands Mueller et al. [2018]. Notably, the recent system Lombardi et al. [2018a] has achieved very impressive results for neural face rendering, while decomposing view-dependent texture and 3D shape modeling.

Over the last several months, several groups have presented results of neural modeling of full bodies Aberman et al. [2018], Chan et al. [2018], Liu et al. [2018], Wang et al. [2018b]. While the presented results are very impressive, the approaches still require a large amount of training data. They also assume that the test images are rendered with the same camera views as the training data, which in our experience makes the task considerably simpler than modeling body appearance from an arbitrary viewpoint. In this work, we aim to expand the neural body modeling approach to tackle the latter, harder task. The work Martin-Brualla et al. [2018] uses a combination of classical and neural rendering to render human body from new viewpoints, but does so based on depth scans and therefore with a rather different algorithmic approach.

A number of recent works **warp a photo of a person** to a new photorealistic image with modified gaze direction Ganin et al. [2016], modified facial expression/pose Cao et al. [2018a], Shu et al. [2018], Tulyakov et al. [2018], Wiles et al. [2018], or modified body pose Balakrishnan et al. [2018], Neverova et al. [2018], Siarohin et al. [2018], Tulyakov et al. [2018], whereas the warping field is estimated using a deep convolutional network (while the original photo effectively serves as a texture). These approaches are however limited in their realism and/or the amount of change they can model, due to their reliance on a single photo of a given person for its input. Our approach also disentangles texture from surface geometry/motion modeling but trains from videos, therefore being able to handle harder problem (full body multi-view setting) and to achieve higher realism.

Our system relies on the **DensePose** body surface parameterization (UV parameterization) similar to the one used in the classical graphics-based representation. Part of our system performs a mapping from the body pose to the surface parameters (UV coordinates) of image pixels. This makes our approach related to the DensePose approach Güler et al. [2018] and the earlier works Güler et al. [2017], Taylor et al. [2012] that predict UV coordinates of image pixels from the input photograph. Furthermore, our approach uses DensePose results Güler et al. [2018] for pretraining.

Our system is related to approaches that extract **textures from multi-view image collections** Goldlücke and Cremers [2009], Lempitsky and Ivanov [2007] or multi-view video collections Volino et al. [2014] or a single video Rav-Acha et al. [2008]. Our approach is also related

to free-viewpoint video compression and rendering systems, e.g. Casas et al. [2014], Collet et al. [2015], Dou et al. [2017], Volino et al. [2014]. Unlike those works, ours is restricted to scenes containing a single human. At the same time, our approach aims to generalize not only to new camera views but also to new user poses unseen in the training videos. The work of Xu et al. [2011] is the most related to ours in this group, as they warp the individual frames of the multi-view video dataset according to the target pose to generate new sequences. The poses that they can handle, however, are limited by the need to have a close match in the training set, which is a strong limitation given the combinatorial nature of the human pose configuration space.

3.3 Method

3.3.1 Notation.

We use the lower index i to denote objects that are specific to the i-th training or test image. We use uppercase notation, e.g. B_i to denote a stack of maps (a third-order tensor/three-dimensional array) corresponding to the i-th training or test image. We use the upper index to denote a specific map (channel) in the stack, e.g. B_i^j. Furthermore, we use square brackets to denote elements corresponding to a specific image location, e.g. $B_i^j[x, y]$ denotes the scalar element in the j-th map of the stack B_i located at location (x, y), and $B_i[x, y]$ denotes the vector of elements corresponding to all maps sampled at location (x, y).

3.3.2 Input and output.

In general, we are interested in synthesizing images of a certain person given her/his pose. We assume that the pose for the i-th image comes in the form of 3D joint positions defined in the camera coordinate frame. As an input to the network, we then consider a map stack B_i, where each map B_i^j contains the rasterized j-th segment (bone) of the "stickman" (skeleton) projected on the camera plane. To retain the information about the third coordinate of the joints, we linearly interpolate the depth value between the joints defining the segments, and use the interpolated values to define the values in the map B_i^j corresponding to the bone pixels (the pixels not covered by the j-th bone are set to zero). Overall, the stack B_i incorporates the information about the person and the camera pose.

As an output of the whole system, we expect an RGB image (a three-channel stack) I_i and a single channel mask M_i, defining the pixels that are covered by the avatar. Below, we consider two approaches: the *direct translation* baseline, which directly maps B_i into $\{I_i, M_i\}$ and the *textured neural avatar* approach that performs such mapping indirectly using texture mapping.

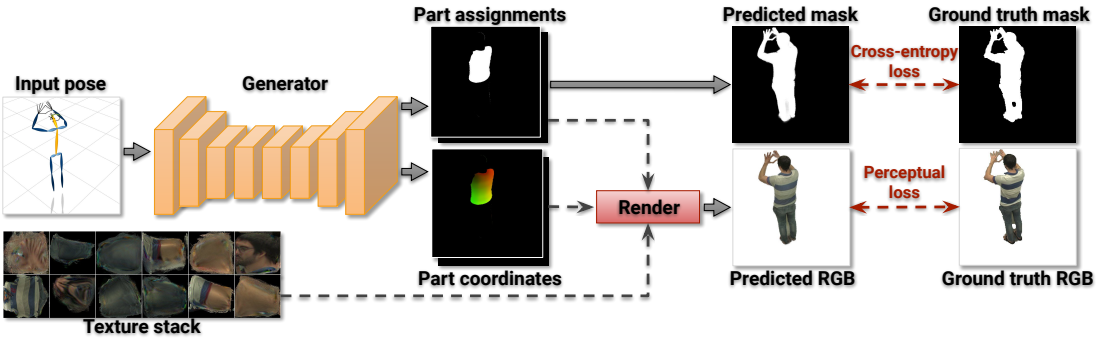


FIGURE 3.1: The overview of the textured neural avatar system. The input pose is defined as a stack of "bone" rasterizations (one bone per channel; here we show it as a skeleton image). The input is processed by the fully-convolutional network (generator) to produce the body part assignment map stack and the body part coordinate map stack. These stacks are then used to sample the body texture maps at the locations prescribed by the part coordinate stack with the weights prescribed by the part assignment stack to produce the RGB image. In addition, the last body assignment stack map corresponds to the background probability. During learning, the mask and the RGB image are compared with ground-truth and the resulting losses are backpropagated through the sampling operation into the fully-convolutional network and onto the texture, resulting in their updates.

In both cases, at training time, we assume that for each input frame i, the input joint locations and the "ground truth" foreground mask are estimated, and we use 3D body pose estimation and human semantic segmentation to extract them from raw video frames. At test time, given a real or synthetic background image $\tilde{I}_i$, we generate the final view by first predicting M_i and I_i from the body pose and then linearly blending the resulting avatar into an image: $\hat{I}_i = I_i \odot M_i + \tilde{I}_i \odot (1 - M_i)$ (where $\odot$ defines a "location-wise" product, i.e. the RGB values at each location are multiplied by the mask value at this location).

3.3.3 Direct translation baseline.

The direct approach that we consider as a baseline to ours is to learn an image translation network that maps the map stack B_i^k to the map stacks I_i and M_i (usually the two output stacks are produced within two branches that share the initial stage of the processing Dosovitskiy et al. [2015]). Generally, mappings between stacks of maps can be implemented using fully-convolutional architectures. Exact architectures and losses for such networks is an active area of research Chen and Koltun [2017], Isola et al. [2017], Johnson et al. [2016a], Ulyanov et al. [2016]. Very recent works Aberman et al. [2018], Chan et al. [2018], Liu et al. [2018], Wang et al. [2018b] have used direct translation (with various modifications) to synthesize the view of a person for a fixed camera. We use the video-to-video variant of this approach Wang et al. [2018b] as a baseline for our method.

3.3.4 Textured neural avatar.

The direct translation approach relies on the generalization ability of ConvNets and incorporates very little domain-specific knowledge into the system. As an alternative, we suggest the textured avatar approach, that explicitly estimates the textures of body parts, thus ensuring the similarity of the body surface appearance under varying pose and cameras.

Following the DensePose approach Güler et al. [2018], we subdivide the body into n=24 parts, where each part has a 2D parameterization. Each body part also has the texture map T^k, which is a color image of a fixed pre-defined size (256×256 in our implementation). The training process for the textured neural avatar estimates personalized part parameterizations and textures.

Again, following the DensePose approach, we assume that each pixel in an image of a person is (soft)-assigned to one of n parts or to the background and with a specific location on the texture of that part (body part coordinates). Unlike DensePose, where part assignments and body part coordinates are induced from the image, our approach at test time aims to predict them based solely on the pose B_i.

The introduction of the body surface parameterization outlined above changes the translation problem. For a given pose defined by B_i, the translation network now has to predict the stack P_i of body part assignments and the stack C_i of body part coordinates, where P_i contains $n+1$ maps of non-negative numbers that sum to identity (i.e. $\sum_{k=0}^{n} P_i^k[x,y] = 1$ for any position (x,y)), and C_i contains $2n$ maps of real numbers between 0 and w, where w is the spatial size (width and height) of the texture maps T^k.

The map channel P_i^k for $k = 0, \ldots, n-1$ is then interpreted as the probability of the pixel to belong to the k-th body part, and the map channel P_i^n corresponds to the probability of the background. The coordinate maps C_i^{2k} and C_i^{2k+1} correspond to the pixel coordinates on the k-th body part. Specifically, once the part assignments P_i and body part coordinates C_i are predicted, the image I_i at each pixel (x,y) is reconstructed as a weighted combination of texture elements, where the weights and texture coordinates are prescribed by the part assignment maps and the coordinate maps correspondingly:

$$s(P_i, C_i, T)[x,y] = \sum_{k=0}^{n-1} P_i^k[x,y] \cdot T^k\left[C_i^{2k}[x,y], C_i^{2k+1}[x,y]\right], \tag{3.1}$$

where $s(\cdot,\cdot,\cdot)$ is the sampling function (layer) that outputs the RGB map stack given the three input arguments. In (3.1), the texture maps T^k are sampled at non-integer locations $(C_i^{2k}[x,y], C_i^{2k+1}[x,y])$ in a piecewise-differentiable manner using bilinear interpolation Jaderberg et al. [2015].

When training the neural textured avatar, we learn a convolutional network g_ϕ with learnable parameters ϕ to translate the input map stacks B_i into the body part assignments and the body part coordinates. As g_ϕ has two branches ("heads"), we denote with g_ϕ^P the branch that produces the body part assignments stack, and with g_ϕ^C the branch that produces the body part coordinates. To learn the parameters of the textured neural avatar, we optimize the loss between the generated image and the ground truth image $\bar{I}_i$:

$$\mathcal{L}_{\text{image}}(\phi, T) = d_{\text{Image}}\left(\bar{I}_i,\ s\left(g_\phi^P(B_i), g_\phi^C(B_i), T\right)\right) \tag{3.2}$$

where $d_{\text{Image}}(\cdot,\cdot)$ is a loss used to compare two images. In our current implementation we use a simple perceptual loss Gatys et al. [2016], Johnson et al. [2016a], Ulyanov et al. [2016], which computes the maps of activations within pretrained fixed VGG network Simonyan and Zisserman [2014] for both images and evaluates the L1-norm between the resulting maps (`Conv1,6,11,20,29` of VGG19 were used). More advanced adversarial losses Goodfellow et al. [2014a] popular in image translation Dosovitskiy and Brox [2016], Isola et al. [2017] can also be used here.

During the stochastic optimization, the gradient of the loss (3.2) is backpropagated through (3.1) both into the translation network g_ϕ and onto the texture maps T^k, so that minimizing this loss updates not only the network parameters but also the textures themselves. As an addition, the learning also optimizes the mask loss that measures the discrepancy between the ground truth background mask $1 - \bar{M}_i$ and the background mask prediction:

$$\mathcal{L}_{\text{mask}}(\phi, T) = d_{\text{BCE}}\left(\bar{1} - M_i,\ g_\phi^P(B_i)^n\right) \tag{3.3}$$

where d_{BCE} is the binary cross-entropy loss, and $g_\phi^P(B_i)^n$ corresponds to the n-th (i.e. background) channel of the predicted part assignment map stack. After backpropagation of the weighted combination of (3.2) and (3.3), the network parameters ϕ and the textures maps T^k are updated. As the training progresses, the texture maps change (Figure 3.1), and so does the body part coordinate predictions, so that the learning is free to choose the appropriate parameterization of body part surfaces.

3.3.5 Initialization of textured neural avatar.

The success of our network depends on the initialization strategy. When training from multiple video sequences, we use the DensePose system Güler et al. [2018] to initialize the textured neural avatar. Specifically, we run DensePose on the training data and pretrain g_ϕ as a translation network between the pose stacks B_i and the DensePose outputs.

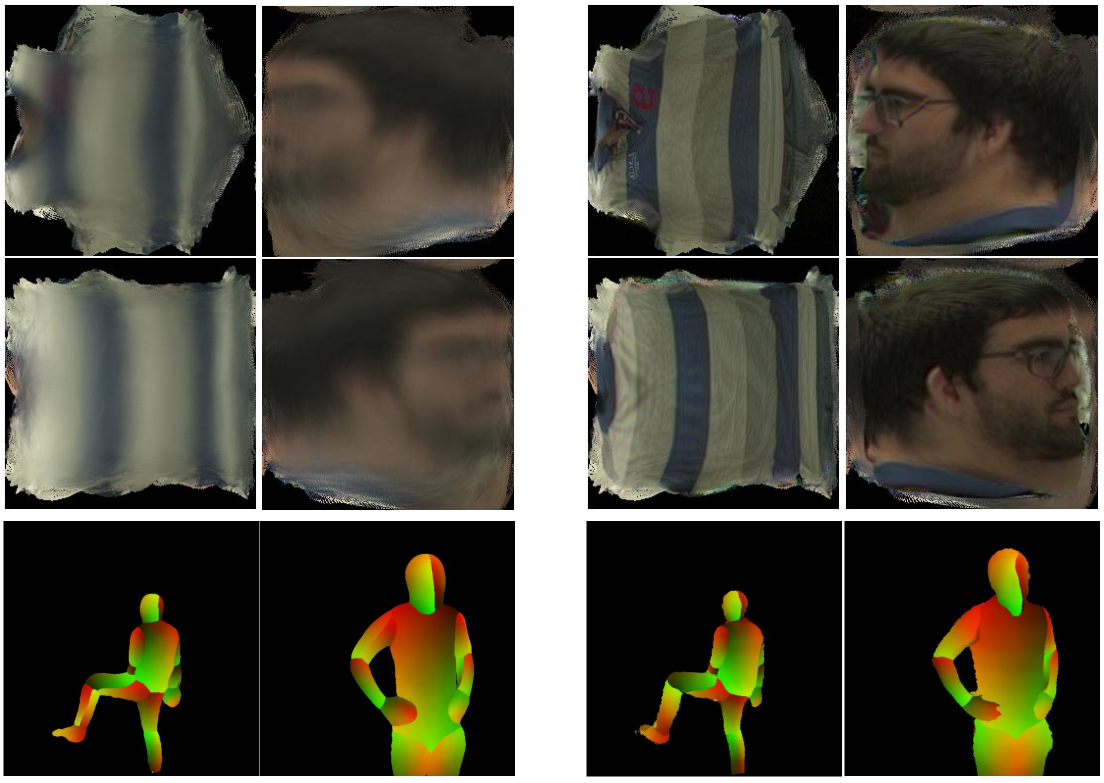


FIGURE 3.2: The impact of the learning on the texture (top, shown for the same subset of maps T^k) and on the convolutional network g_ϕ^C predictions (bottom, shown for the same pair of input poses). Left part shows the starting state (after initialization), while the right part shows the final state, which is considerably different from the start.

An alternative way that is particularly attractive when training data is scarce is to initialize the avatar is through transfer learning. In this case, we simply take g_ϕ from another avatar trained on abundant data. The explicit decoupling of geometry from appearance in our method facilitates transfer learning, as the geometrical mapping provided by the network g_ϕ usually does not need to change much between two people, especially if the body types are not too dissimilar.

Once the mapping g_ϕ has been initialized, the texture maps T^k are initialized as follows. Each pixel in the training image is assigned to a single body part (according to the prediction of the pretrained g_ϕ^P) and to a particular texture pixel on the texture of the corresponding part (according to the prediction of the pretrained g_ϕ^C). Then, the value of each texture pixel is initialized to the mean of all image pixels assigned to it (the texture pixels assigned zero pixels are initialized to black). The initialized texture T and g_ϕ usually produce images that are only coarsely reminding the person, and they change significantly during the end-to-end learning (Figure 3.2).

FIGURE 3.3: Renderings produced by multiple textured neural avatars (for all people in our study). All renderings are produced from the new viewpoints unseen during training.

	(a) User study		(b) SSIM score			(c) Frechet distance		
	Ours-v-V2V	Ours-v-Direct	V2V	Direct	Ours	V2V	Direct	Ours
CMU1-16	0.56	0.75	0.908	0.899	0.919	6.7	7.3	8.8
CMU2-16	0.54	0.74	0.916	0.907	0.922	7.0	8.8	10.7
CMU1-6	0.50	0.92	0.905	0.896	0.914	7.7	10.7	8.9
CMU2-6	0.53	0.71	0.918	0.907	0.920	7.0	9.7	10.4

TABLE 3.1: Quantitative comparison of the three models operating on different datasets (see text for discussion).

3.3.6 Experiments

Below, we discuss the details of the experimental validation, provide comparison with baseline approaches, and show qualitative results. The project webpage[1] contains more videos of the learned avatars.

3.3.6.1 Architecture.

We input 3D pose via bone rasterizations, where each bone, hand and face are drawn in separate channels. We then use standard image translation architecture Johnson et al. [2016a] to perform a mapping from these bones' rasterizations to texture assignments and coordinates. This architecture consists of downsampling layers, stack of residual blocks, operating at low dimensional feature representations, and upsampling layers. We then split the network into two roughly equal parts: encoder and decoder, with texture assignments and coordinates having separate decoders. We use 4 downsampling and upsampling layers with initial 32 channels in the convolutions and 256 channels in the residual blocks. The ConvNet g_ϕ has 17 million parameters.

[1] https://saic-violet.github.io/texturedavatar/

3.3.6.2 Datasets.

We train neural avatars on several types of datasets. First, we consider collections of multi-view videos registered in time and space, where 3D pose estimates can be obtained via triangulation of 2D poses. We use two subsets (corresponding to two persons from the 171026_pose2 scene) from the CMU Panoptic dataset collection Joo et al. [2017], referring to them as `CMU1` and `CMU2` (both subsets have approximately four minutes / 7,200 frames in each camera view). We consider two regimes: training on 16 cameras (`CMU1-16` and `CMU2-16`) or six cameras (`CMU1-6` and `CMU2-6`). The evaluation is done on the hold-out cameras and hold-out parts of the sequence (no overlap between train and test in terms of the cameras or body motion).

We have also captured our own multi-view sequences of three subjects using a rig of seven cameras, spanning approximately 30°. In one scenario, the training sets included six out of seven cameras, where the duration of each video was approximately six minutes (11,000 frames). We show qualitative results for the hold-out camera as well as from new viewpoints. In the other scenario described below, training was done based on a video from a single camera.

Finally, we evaluate on two short monocular sequences from Alldieck et al. [2018b] and a Youtube video in Figure 3.6.

3.3.6.3 Pre-processing.

Our system expects 3D human pose as input. For non-CMU datasets, we used the OpenPose-compatible Cao et al. [2017], Simon et al. [2017] 3D pose formats, represented by 25 body joints, 21 joints for each hand and 70 facial landmarks. For the CMU Panoptic datasets, we use the available 3D pose annotation as input (which has 19 rather than 25 body joints). To get a 3D pose for non-CMU sequences we first apply the OpenPose 2D pose estimation engine to five consecutive frames of the monocular RGB image sequence. Then we concatenate and lift the estimated 2D poses to infer the 3D pose of the last frame by using a multi-layer perceptron model. The perceptron is trained on the CMU 3D pose annotations (augmented with position of the feet joints by triangulating the output of OpenPose) in orthogonal projection.

For foreground segmentation we use DeepLabv3+ with Xception-65 backbone Chen et al. [2018] initially trained on PASCAL VOC 2012 Everingham et al. [2015] and fine-tuned on HumanParsing dataset Liang et al. [2015a,b] to predict initial human body segmentation masks. We additionally employ GrabCut Rother et al. [2004] with background/foreground model initialized by the masks to refine object boundaries on the high-resolution images. Pixels covered by the skeleton rasterization were always added to the foreground mask.

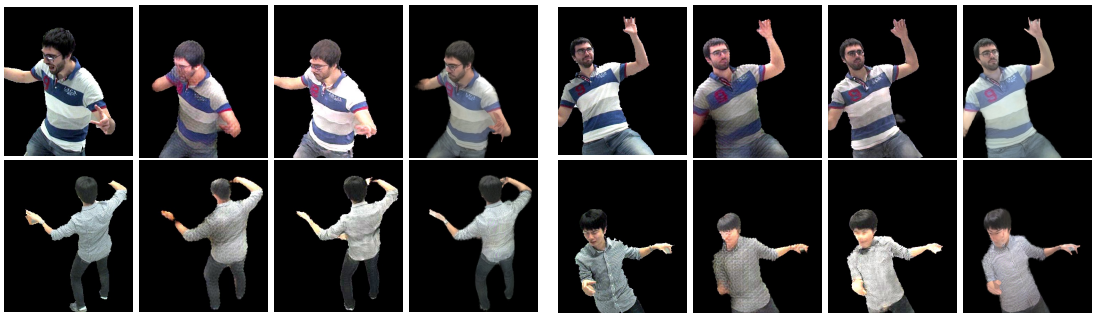


FIGURE 3.4: Comparison of the rendering quality for the Direct, V2V and proposed methods on the `CMU1-6` and `CMU2-6` sequences. Images from six arbitrarily chosen cameras were used for training. We generate the views onto the hold-out cameras which were not used during training. The pose and camera in the lower right corner are in particular difficult for all the systems.

3.3.6.4 Baselines.

In the multi-video training scenario, we consider two other systems, against which ours is compared. First, we take the video-to-video (*V2V*) system Wang et al. [2018b], using the authors' code with minimal modifications that lead to improved performance. We provide it with the same input as ours, and we use images with blacked-out background (according to our segmentation) as desired output. On the `CMU1-6` task, we have also evaluated a model with DensePose results computed on the target frame given as input (alongside keypoints). Despite much stronger (oracle-type) conditioning, the performance of this model in terms of considered metrics has not improved in comparison with V2V that uses only body joints as input.

The video-to-video system employs several adversarial losses and an architecture different from ours. Therefore we consider a more direct ablation (*Direct*), which has the same network architecture that predicts RGB color and mask directly, rather than via body part assignments/coordinates. The Direct system is trained using the same losses and in the same protocol as ours.

As for the single video case, two baseline systems, against which ours is compared, were considered. On our own captured sequences, we compare our system against video-to-video (*V2V*) system Wang et al. [2018b], whereas on sequences from Alldieck et al. [2018b] we provide a qualitative comparison against the system of Alldieck et al. [2018b].

3.3.6.5 Multi-video comparison.

We compare the three systems (*ours*, *V2V*, *Direct*) in `CMU1-16`, `CMU2-16`, `CMU1-6`, `CMU2-6`. Using the hold-out sequences/motions, we then evaluated two popular metrics, namely structured self-similarity (SSIM) and Frechet Inception Distance (FID) between the results of each system and the hold-out frames (with background removed using our segmentation algorithm). Our

method outperforms the other two in terms of SSIM and underperforms V2V in terms of FID. Representative examples are shown in Figure 3.4.

We have also performed user study using a crowd-sourcing website, where the users were shown the results of ours and one of the other two systems on either side of the ground truth image and were asked to pick a better match to the middle image. In the side-by-side comparison, the results of our method were always preferred by the majority of crowd-sourcing users. We note that our method suffers from a disadvantage both in the quantitative metrics and in the user comparison, since it averages out lighting from different viewpoints. The more detailed quantitative comparison is presented in Table 3.1.

We show more qualitative examples of our method for a variety of models in Figure 3.3 and some qualitative comparisons with baselines in Figure 3.5.

3.3.6.6 Single video comparisons.

We also evaluate our system in a single video case. We consider the scenario, where we train the model and transfer it to a new person by fitting it to a single video. We use single-camera videos from one of the cameras in our rig. We then evaluate the model (and V2V baseline) on a hold-out set of poses projected onto the camera from the other side of the rig (around 30° away). We thus demonstrate that new models can be obtained using a single monocular video. For our models, we consider transferring from `CMU1-16`.

We thus pretrain V2V and our system on `CMU1-16` and use the obtained weights of g_ϕ as initialization for fine-tuning to the single video in our dataset. The texture maps are initialized from scratch as described above. Evaluating on hold-out camera and motion highlighted strong advantage of our method. In the user study on two subjects, the result of our method has been preferred to V2V in 55% and 65% of the cases. We further compare our method and the system of Alldieck et al. [2018b] on the sequences from Alldieck et al. [2018b]. The qualitative comparison is shown in Figure 3.6. In addition, we generate an avatar from a YouTube video. In this set of experiments, the avatars were obtained by fine-tuning from the same avatar (shown in Figure 3.5–left). Except for the considerable artefacts on hand parts, our system has generated avatars that can generalize to new pose despite very short video input (300 frames in the case of Alldieck et al. [2018b]).

3.3.6.7 Limitations

Our method suffers from certain limitations. The generalization ability is still limited, as it does not generalize well when a person is rendered at a scale that is considerably different from

FIGURE 3.5: Results comparison for our multi-view sequences using a hold-out camera. Textured Neural Avatars and the images produced by the video-to-video (V2V) system correspond to the same viewpoint. Both systems use a video from a single viewpoint for training. *Electronic zoom-in recommended.*

FIGURE 3.6: Results on external monocular sequences. Rows 1-2: avatars for sequences from Alldieck et al. [2018b] in an unseen pose (left – ours, right – Alldieck et al. [2018b]). Row 3 – the textured avatar computed from a popular YouTube video ('PUMPED UP KICKS DUBSTEP'). In general, our system is capable of learning avatars from monocular videos.

the training set (which can be partially addressed by rescaling prior to rendering followed by cropping/padding postprocessing). Furthermore, textured avatars exhibit strong artefacts in the presence of pose estimation errors on hands and faces. Finally, our method assumes constancy of the surface color and ignores lighting effects. This can be potentially addressed by making our textures view- and lighting-dependent Debevec et al. [1998], Lombardi et al. [2018a].

3.4 Summary and Discussion

We have presented textured neural avatar approach to model the appearance of humans for new camera views and new body poses. Our system takes the middle path between the recent generation of methods that use ConvNets to map the pose to the image directly, and the traditional approach that uses geometric modeling of the surface and superimpose the personalized texture maps. This is achieved by learning a ConvNet that predicts texture coordinates of pixels in the new view jointly with the texture within the end-to-end learning process. We demonstrate that retaining an explicit shape and texture separation helps to achieve better generalization than direct mapping approaches.

Our method suffers from certain limitations. The generalization ability is still limited, as it does not generalize well when a person is rendered at a scale that is considerably different from the training set (which can be partially addressed by rescaling prior to rendering followed by cropping/padding postprocessing). Furthermore, textured avatars exhibit strong artefacts in the presence of pose estimation errors on hands and faces. Finally, our method assumes constancy of the surface color and ignores lighting effects. This can be potentially addressed by making our textures view- and lighting-dependent Debevec et al. [1998], Lombardi et al. [2018a].

Chapter 4

Few-Shot Adversarial Learning of Realistic Neural Talking Head Models

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.

This work was published as: Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. *Few-Shot Adversarial Learning of Realistic Neural Talking Head Models.* International Conference on Computer Vision (ICCV), 2019.

Supplementary materials are hosted on the project page: https://samsunglabs.github.io/talking_heads

4.1 Introduction

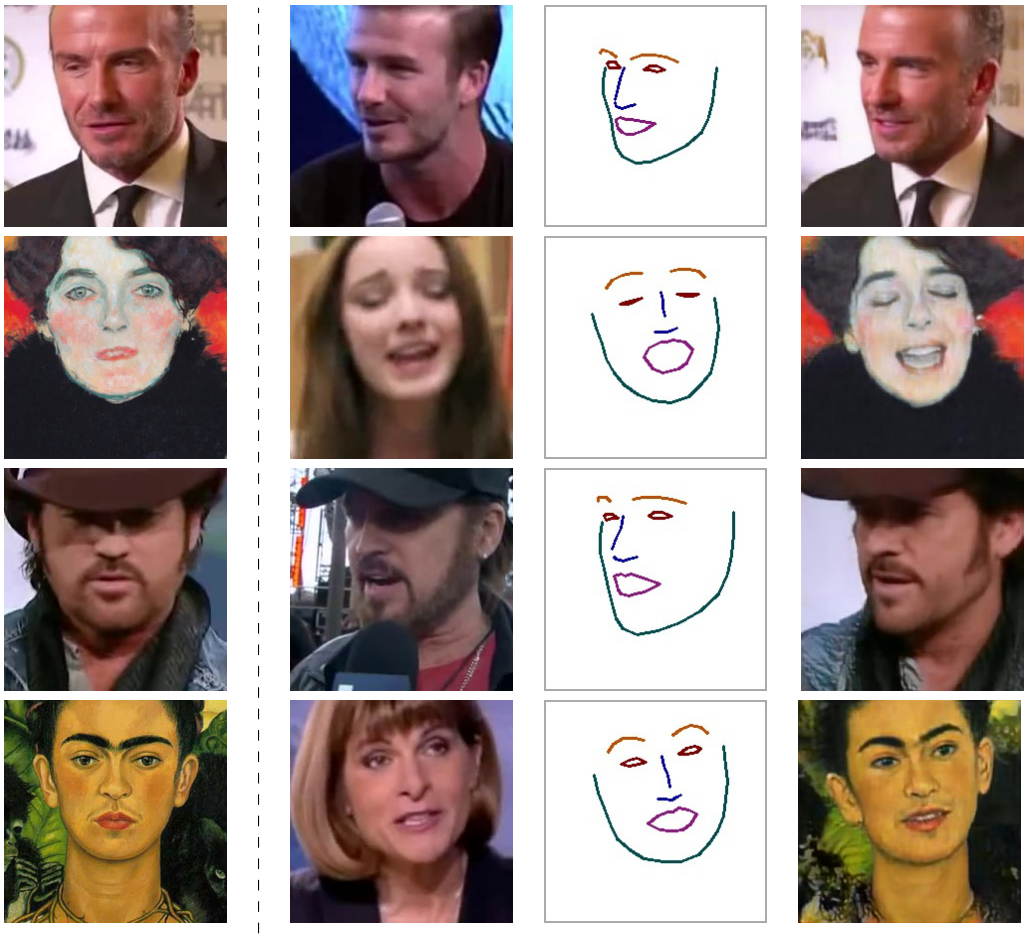


FIGURE 4.1: The results of talking head image synthesis using face landmark tracks extracted from a different video sequence of the same person (on the left), and using face landmarks of a different person (on the right). The **results** are conditioned on the **landmarks** taken from the **target** 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.

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 video-conferencing 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 Mori [1970]). 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 Averbuch-Elor et al. [2017], Seitz and Dyer [1996] and warping fields synthesized using machine learning (including deep learning) Ganin et al. [2016], Shu et al. [2018], Wiles et al. [2018] 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 Isola et al. [2017], Kim et al. [2018a], Wang et al. [2018c]. 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 Kim et al. [2018a], Wang et al. [2018c] or a large dataset of photographs Isola et al. [2017] 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 Alexander et al. [2010b], 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 Isola et al. [2017], Kim et al. [2018a], Wang et al. [2018c], 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 Isola et al. [2017], Wiles et al. [2018] 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).

4.2 Related work

4.2.1 Systems based on statistical shape modeling

A huge body of works is devoted to statistical modeling of the appearance of human faces Blanz et al. [1999], with remarkably good results obtained both with classical techniques Thies et al. [2016b] and, more recently, with deep learning Lombardi et al. [2018a], Nagano et al. [2019] (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 Thies et al. [2016b] or lips modeling Suwajanakorn et al. [2017] 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.

4.2.2 Generative adversarial networks

The design of our system borrows a lot from the recent progress in generative modeling of images. Thus, our architecture uses adversarial training Goodfellow et al. [2014a] and, more specifically, the ideas behind conditional discriminators Mehdi Mirza [2014], including projection discriminators Takeru Miyato [2018]. Our meta-learning stage uses the adaptive instance normalization mechanism Huang and Belongie [2017], which was shown to be useful in large-scale conditional generation tasks Brock et al. [2019a], Tero Karras [2018]. We also find an idea of content-style decomposition Huang et al. [2018] to be extremely useful for separating the texture from the body pose.

4.2.3 Few-shot training via meta-learning

The model-agnostic meta-learner (MAML) Finn et al. [2017] 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 Antoniou et al. [2018], MetaGAN Zhang et al. [2018b], adversarial meta-learning Yin et al. [2018] 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 Arik et al. [2018], Jia et al. [2018]. 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.

4.3 Methods

4.3.1 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 Bulat and Tzimiropoulos [2017a] 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 4.2):

- 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

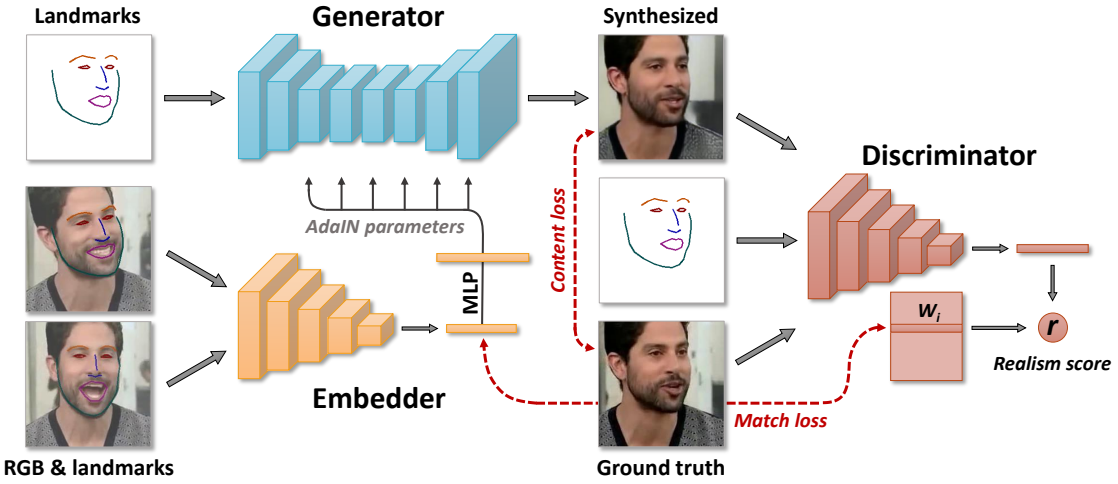


FIGURE 4.2: Our meta-learning architecture involves the embedder network that maps head images (with estimated face landmarks) to the embedding vectors, which contain pose-independent information. The generator network maps input face landmarks into output frames through the set of convolutional layers, 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.

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 $\mathbf{W}, \mathbf{w}_0, b$.

4.3.2 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:

$$\hat{\mathbf{e}}_i = \frac{1}{K}\sum_{k=1}^{K} E\left(\mathbf{x}_i(s_k), \mathbf{y}_i(s_k);\, \phi\right) . \tag{4.1}$$

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

$$\hat{\mathbf{x}}_i(t) = G\left(\mathbf{y}_i(t), \hat{\mathbf{e}}_i;\, \psi, \mathbf{P}\right) . \tag{4.2}$$

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:

$$\begin{aligned}\mathcal{L}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \mathcal{L}_{\text{CNT}}(\phi, \psi, \mathbf{P}) + \\ \mathcal{L}_{\text{ADV}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) + \mathcal{L}_{\text{MCH}}(\phi, \mathbf{W}) .\end{aligned} \tag{4.3}$$

In (4.3), the content loss term $\mathcal{L}_{\text{CNT}}$ 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 Johnson et al. [2016a], corresponding to VGG19 Simonyan and Zisserman [2015a] network trained for ILSVRC classification and VGGFace Parkhi et al. [2015a] 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 (4.3) corresponds to the realism score computed by the discriminator, which needs to be maximized, and a feature matching term Wang et al. [2018e], which essentially is a perceptual similarity measure, computed using discriminator (it helps with the stability of the training):

$$\begin{aligned}\mathcal{L}_{\text{ADV}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \\ -D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b) + \mathcal{L}_{\text{FM}} .\end{aligned} \tag{4.4}$$

Following the projection discriminator idea Takeru Miyato [2018], 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:

$$\begin{aligned}D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i;\, \theta, \mathbf{W}, \mathbf{w}_0, b) = \\ V\left(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t);\, \theta\right)^T (\mathbf{W}_i + \mathbf{w}_0) + b ,\end{aligned} \tag{4.5}$$

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 $\mathcal{L}_{\text{MCH}}(\phi, \mathbf{W})$ in (4.3) 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 $\hat{\mathbf{x}}_i(t)$:

$$\begin{aligned}\mathcal{L}_{\text{DSC}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = \\ \max(0, 1 + D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i;\, \phi, \psi, \theta, \mathbf{W}, \mathbf{w}_0, b)) + \\ \max(0, 1 - D(\mathbf{x}_i(t), \mathbf{y}_i(t), i;\, \theta, \mathbf{W}, \mathbf{w}_0, b))\,.\end{aligned} \tag{4.6}$$

The objective (4.6) 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 $\mathcal{L}_{\text{CNT}}, \mathcal{L}_{\text{ADV}}$ and $\mathcal{L}_{\text{MCH}}$ with the updates of the discriminator that minimize the loss $\mathcal{L}_{\text{DSC}}$.

4.3.3 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:

$$\hat{\mathbf{e}}_{\text{NEW}} = \frac{1}{T}\sum_{t=1}^{T} E(\mathbf{x}(t), \mathbf{y}(t);\, \phi)\,, \tag{4.7}$$

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

$$D'(\hat{\mathbf{x}}(t), \mathbf{y}(t);\, \theta, \mathbf{w}', b) = V\left(\hat{\mathbf{x}}(t), \mathbf{y}(t);\, \theta\right)^T \mathbf{w}' + b\,. \tag{4.8}$$

As can be seen from the comparison of expressions (4.5) and (4.8), 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:

$$\mathcal{L}'(\psi, \psi', \theta, \mathbf{w}', b) = \mathcal{L}'_{\text{CNT}}(\psi, \psi') + \mathcal{L}'_{\text{ADV}}(\psi, \psi', \theta, \mathbf{w}', b)\,, \tag{4.9}$$

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 (4.6):

$$\begin{aligned} &\mathcal{L}'_{\text{DSC}}(\psi, \psi', \theta, \mathbf{w}', b) = \\ &\max(0, 1 + D(\hat{\mathbf{x}}(t), \mathbf{y}(t);\, \psi, \psi', \theta, \mathbf{w}', b)) + \\ &\qquad \max(0, 1 - D(\mathbf{x}(t), \mathbf{y}(t);\, \theta, \mathbf{w}', b))\,. \end{aligned} \tag{4.10}$$

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.

4.3.4 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. Johnson et al. [2016a], but replace downsampling and upsampling layers with residual blocks similarly to Brock et al. [2019a] (with batch normalization Ioffe and Szegedy [2015] replaced by instance normalization Ulyanov et al. [2016]). The person-specific parameters $\hat{\psi}_i$ serve as the affine coefficients of instance normalization layers, following the adaptive instance normalization technique proposed in Huang and Belongie [2017], 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 Takeru et al. [2018] for all convolutional and fully connected layers in all the networks. We also use self-attention blocks, following Brock et al. [2019a] and Zhang et al. [2019]. 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.

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

We use Caffe Jia et al. [2014] 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 Kingma and Ba [2014]. 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 Zhang et al. [2019].

4.4 Experiments

Two datasets with talking head videos are used for quantitative and qualitative evaluation: VoxCeleb1 Nagrani et al. [2017] (256p videos at 1 fps) and VoxCeleb2 Chung et al. [2018a] (224p videos at 25 fps), with the latter having approximately 10 times more videos 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.

4.4.1 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) Heusel et al. [2017a], mostly measuring perceptual realism, structured similarity (SSIM) Wang et al. [2004a], 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 Deng et al. [2019] 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.

4.4.2 Methods.

On the VoxCeleb1 dataset we compare our model against two other systems: X2Face Wiles et al. [2018] and Pix2pixHD Wang et al. [2018e]. 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 VoxCeleb1 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

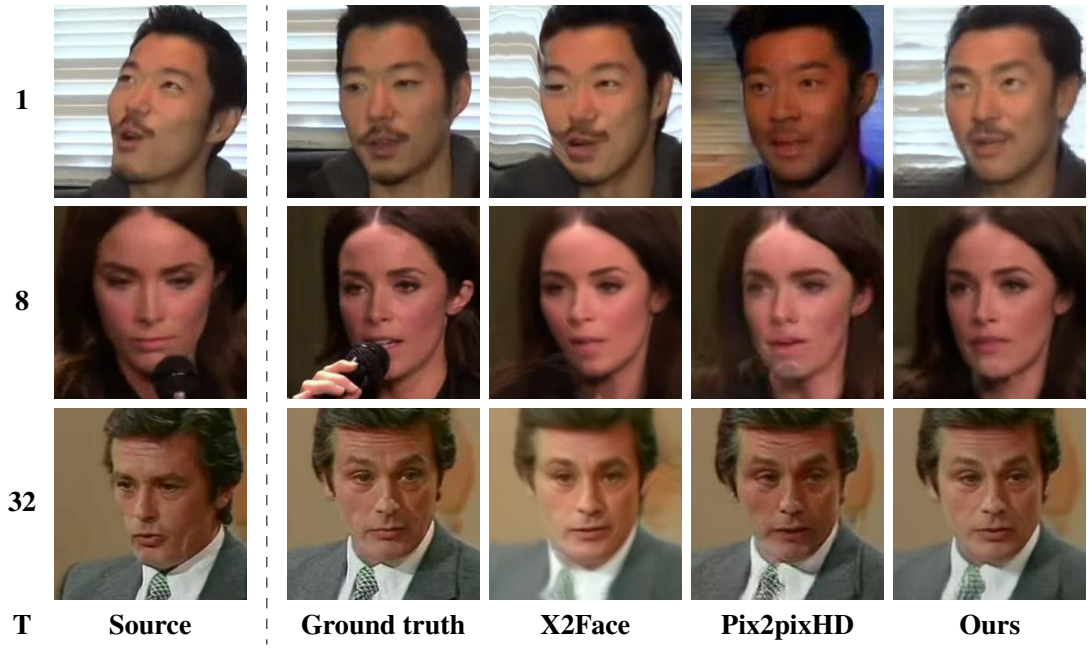


FIGURE 4.3: Comparison on the VoxCeleb1 dataset. For each of the compared methods, we perform one- and few-shot learning on a video of a person not seen during meta-learning or pretraining. We set the number of training frames equal to **T** (the leftmost column). One of the training frames is shown 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.

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.

4.4.3 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 4.1-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 Wiles et al. [2018], 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 4.3 and the

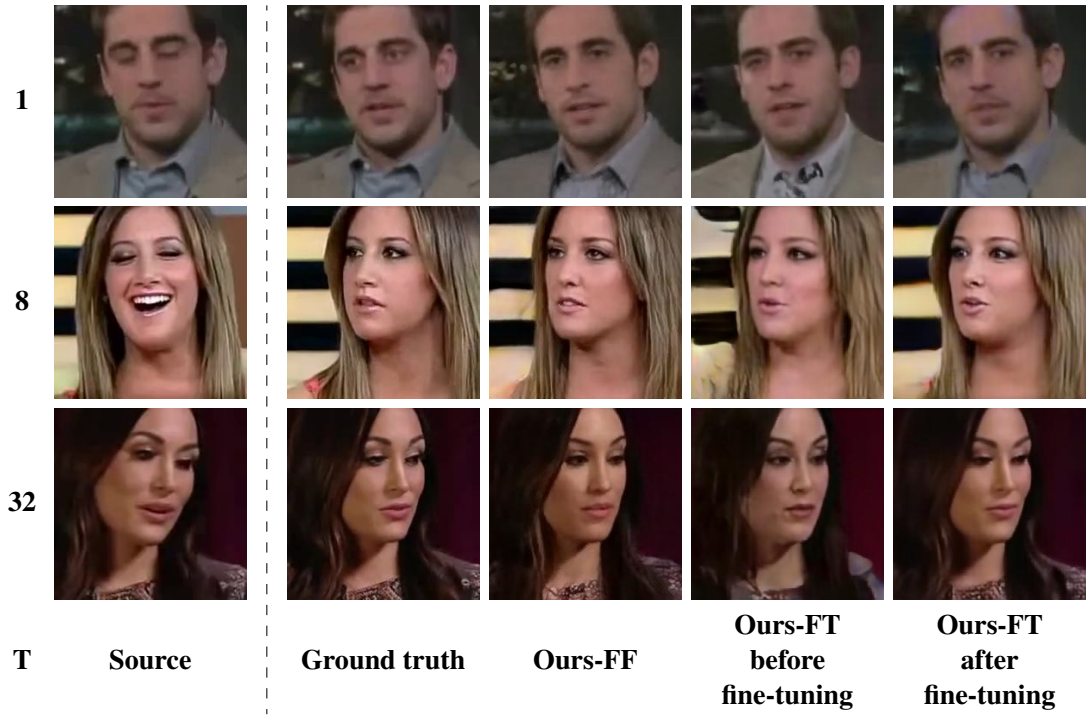


FIGURE 4.4: Results for our best models on the VoxCeleb2 dataset. The number of training frames is, again, equal to **T** (the leftmost column) and the example training frame in shown in the **source** column. Next columns show **ground truth** image and the results for **Ours-FF** feed-forward model, **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.

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 4.1-Top with the results presented in Figure 4.3.

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.

4.4.4 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 $\hat{\mathbf{e}}_{\text{NEW}}$). The FT variant is trained for half as much (75 epochs) but with $\mathcal{L}_{\text{MCH}}$, which allows fine-tuning. We run the

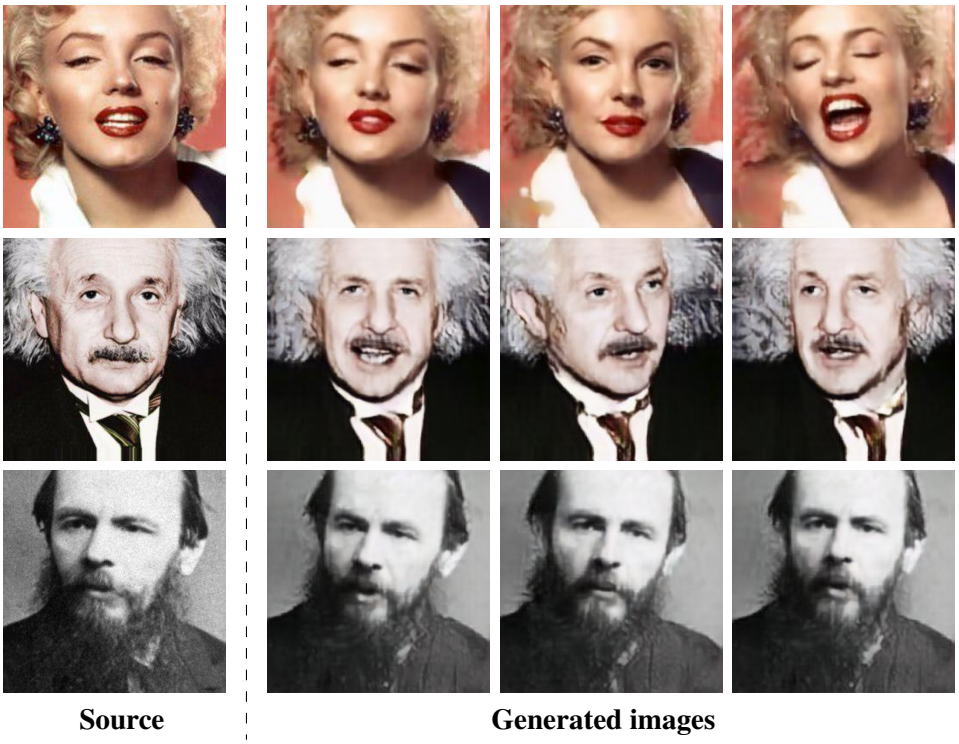


FIGURE 4.5: Bringing still photographs to life. We show the puppeteering results for one-shot models learned from photographs in the **source** column. Driving poses were taken from the VoxCeleb2 dataset. Digital zoom recommended.

evaluation for both of these models since they allow to trade off few-shot learning speed versus the results quality. Both of them achieve considerably 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.

Generally, judging by the results of comparisons (Table 4.1-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.

4.4.5 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 4.5.

4.4.6 Limitations

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.

4.5 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.

Chapter 5

Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars

Abstract

We propose a neural rendering-based system that creates head avatars from a single photograph. Our approach models a person's appearance by decomposing it into two layers. The first layer is a pose-dependent coarse image that is synthesized by a small neural network. The second layer is defined by a pose-independent texture image that contains high-frequency details. The texture image is generated offline, warped and added to the coarse image to ensure a high effective resolution of synthesized head views. We compare our system to analogous state-of-the-art systems in terms of visual quality and speed. The experiments show significant inference speedup over previous neural head avatar models for a given visual quality. We also report on a real-time smartphone-based implementation of our system.

This work was published as: Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. *Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars.* European Conference on Computer Vision (ECCV), 2020.

Supplementary materials are hosted on the project page: https://samsunglabs.github.io/bilayer_model

5.1 Introduction

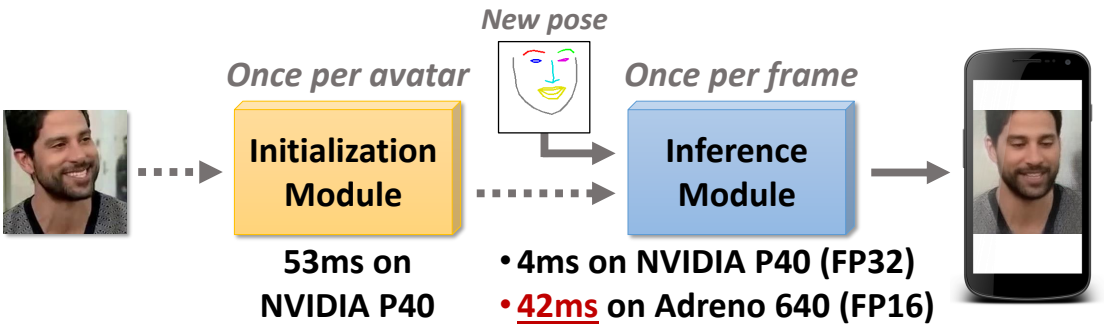


FIGURE 5.1: Our new architecture creates photorealistic neural avatars in one-shot mode and achieves considerable speed-up over previous approaches. Rendering takes just 42 milliseconds on Adreno 640 (Snapdragon 855) GPU, FP16 mode.

Personalized head avatars driven by keypoints or other mimics/pose representation is a technology with manifold applications in telepresence, gaming, AR/VR applications, and special effects industry. Modeling human head appearance is a daunting task, due to complex geometric and photometric properties of human heads including hair, mouth cavity and surrounding clothing. For at least two decades, creating head avatars (talking head models) was done with computer graphics tools using mesh-based surface models and texture maps. The resulting systems fall into two groups. Some Alexander et al. [2010a] are able to model specific people with very high realism after significant acquisition and design efforts are spent on those particular people. Others Hu et al. [2017] are able to create talking head models from as little as a single photograph, but do not aim to achieve photorealism.

In recent years, *neural talking heads* have emerged as an alternative to classic computer graphics pipeline striving to achieve both high realism and ease of acquisition. The first works required a video Kim et al. [2018a], Wang et al. [2018c] or even multiple videos Lombardi et al. [2018a], Suwajanakorn et al. [2017] to create a neural network that can synthesize talking head view of a person. Most recently, several works Fu et al. [2019], Ha et al. [2019], Siarohin et al. [2019c,c], Tripathy et al. [2019], Wang et al. [2019], Zakharov et al. [2019] presented systems that create neural head avatars from a handful of photographs (*few-shot* setting) or a single photograph (*one-shot* setting), causing both excitement and concerns about potential misuse of such technology.

Existing few-shot neural head avatar systems achieve remarkable results. Yet, unlike some of the graphics-based avatars, the neural systems are too slow to be deployed on mobile devices and require a high-end desktop GPU to run in real-time. We note that most application scenarios of neural avatars, especially those related to telepresence, would benefit highly from the capability to run in real-time on a mobile device. While in theory neural architectures within state-of-the-art

approaches can be scaled down in order to run faster, we show that such scaling down results in a very unfavourable speed-realism tradeoff.

In this work, we address the speed limitataions of one-shot neural head avatar systems, and develop an approach that can run much faster than previous models. To achieve this, we adopt a *bi-layer representation*, where the image of an avatar in a new pose is generated by summing two components: a coarse image directly predicted by a rendering network, and a warped texture image. While the warping itself is also predicted by the rendering network, the texture is estimated at the time of avatar creation and is static at runtime. To enable the few-shot capability, we use the meta-learning stage on a dataset of videos, where we (meta)-train the inference (rendering) network, the embedding network, as well as the texture generation network.

The separation of the target frames into two layers allows us both to improve the effective resolution and the speed of neural rendering. This is because we can use off-line avatar generation stage to synthesize high-resolution texture, while at test time both the first component (coarse image) and the warping of the texture need not contain high frequency details and can therefore be predicted by a relatively small rendering network. These advantages of our system are validated by extensive comparisons with previously proposed neural avatar systems. We also report on the smartphone-based real-time implementation of our system, which was beyond the reach of previously proposed models.

5.2 Related work

5.2.1 Few and multi-shot head avatars

As discussed above, methods for the neural synthesis of realistic talking head sequences can be divided into many-shot (i.e. requiring a video or multiple videos of the target person for learning the model) Isola et al. [2017], Kim et al. [2018a], Lombardi et al. [2018a], Wang et al. [2018c] and a more recent group of few-shot/singe-shot methods capable of acquiring the model of a person from a single or a handful photographs Ha et al. [2019], Siarohin et al. [2019c], Tripathy et al. [2019], Wang et al. [2019], Wiles et al. [2018], Zakharov et al. [2019]. Our method falls into the latter category as we focus on the one-shot scenario (modeling from a single photograph).

5.2.2 Direct synthesis approach

Along another dimension, these methods can be divided according to the architecture of the generator network. Thus, several methods Kim et al. [2018a], Tripathy et al. [2019], Wang et al. [2018c], Zakharov et al. [2019] use generators based on *direct synthesis*, where the image

is generated using a sequence of convolutional operators, interleaved with elementwise non-linearities, and normalizations. Person identity information may be injected into such architecture, either with a lengthy learning process (in the many-shot scenario) Kim et al. [2018a], Wang et al. [2018c] or by using adaptive normalizations conditioned on person embeddings Fu et al. [2019], Tripathy et al. [2019], Zakharov et al. [2019]. The method Zakharov et al. [2019] effectively combines both approaches by injecting identity through adaptive normalizations, and then fine-tuning the resulting generator on the few-shot learning set. The direct synthesis approach for human heads can be traced back to Suwajanakorn et al. [2017] that generated lips of a famous person in the talking head sequence, and further towards first works on conditional convolutional neural synthesis of generic objects such as Dosovitskiy et al. [2015].

5.2.3 Differentiable warping approach

The alternative to the direct image synthesis is to use differentiable warping Jaderberg et al. [2015] inside the architecture. The X2Face approach Wiles et al. [2018] applies warping twice, first from the source image to a standardized image (texture), and then to the target image. The Codec Avatar system Lombardi et al. [2018a] synthesizes a pose-dependent texture for a simplified mesh geometry. The MarioNETte system Ha et al. [2019] applies warping to the intermediate feature representations. The Few-shot Vid-to-Vid system Wang et al. [2019] combines direct synthesis with the warping of the previous frame in order to obtain temporal continuity. The First Order Motion Model Siarohin et al. [2019c] learns to warp the intermediate feature representation of the generator based on keypoints that are learned from data. Beyond heads, differentiable warping/texturing have recently been used for full body re-rendering Neverova et al. [2018], Shysheya et al. [2019]. Earlier, DeepWarp system Ganin et al. [2016] used neural warping to alter the appearance of eyes for the purpose of gaze redirection, and Zhou et al. [2016] also used neural warping for the resynthesis of generic scenes. Our method combines direct image synthesis with warping in a new way, as we obtain the fine layer by warping an RGB *pose-independent* texture, while the coarse-grained *pose-dependent* RGB component is synthesized by a neural network directly.

5.3 Methods

We use video sequences annotated with keypoints and, optionally, segmentation masks, for training. We denote t-th frame of the i-th video sequence as $\mathbf{x}^i(t)$, corresponding keypoints as $\mathbf{y}^i(t)$, and segmentation masks as $\mathbf{m}^i(t)$ We will use an index t to denote a target frame, and s – a source frame. Also, we mark all tensors, related to generated images, with a hat symbol, ex. $\hat{\mathbf{x}}^i(t)$. We assume the spatial size of all frames to be constant and denote it as $H \times W$. In some

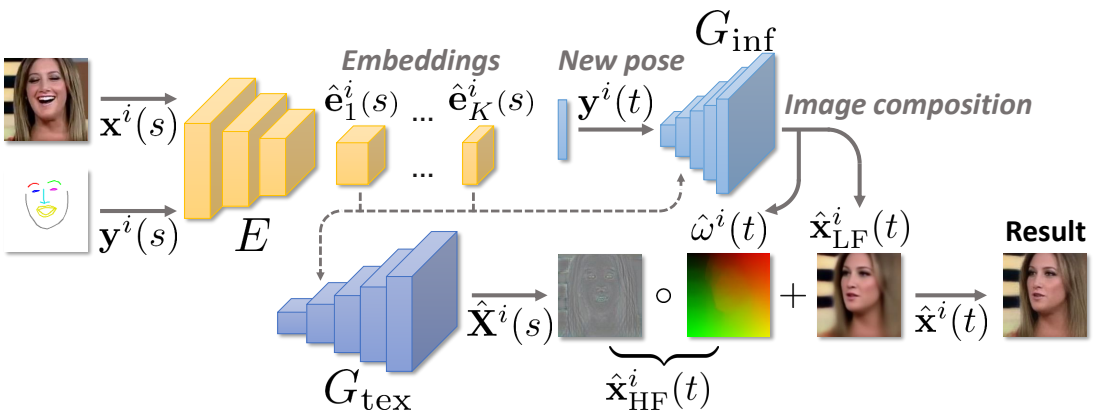


FIGURE 5.2: During training, we first encode a source frame into the embeddings, then we initialize adaptive parameters of both inference and texture generators, and predict a high-frequency texture. These operations are only done once per avatar. Target keypoints are then used to predict a low-frequency component of the output image and a warping field, which, applied to the texture, provides the high-frequency component. Two components are then added together to produce an output.

modules, input keypoints are encoded as an RGB image, which is a standard approach in a large body of previous works Ha et al. [2019], Wang et al. [2019], Zakharov et al. [2019]. In this work, we will call it a *landmark* image. But, contrary to these approaches, at test-time we input the keypoints into the inference generator directly as a vector. This allows us to significantly reduce the inference time of the method.

5.3.1 Architecture

In our approach, the following networks are trained in an end-to-end fashion:

- The *embedder* network $E\big(\mathbf{x}^i(s), \mathbf{y}^i(s)\big)$ encodes a concatenation of a source image and a landmark image into a stack of embeddings $\{\hat{\mathbf{e}}^i_k(s)\}$, which are used for initialization of the adaptive parameters inside the generators.
- The *texture generator* network $G_{\text{tex}}\big(\{\hat{\mathbf{e}}^i_k(s)\}\big)$ initializes its adaptive parameters from the embeddings and decodes an inpainted high-frequency component of the source image, which we call a texture $\hat{\mathbf{X}}^i(s)$.
- The *inference generator* network $G\big(\mathbf{y}^i(t), \{\hat{\mathbf{e}}^i_k(s)\}\big)$ maps target poses into a predicted image $\hat{\mathbf{x}}^i(t)$. The network accepts vector keypoints as an input and outputs a low-frequency layer of the output image $\hat{\mathbf{x}}^i_{\text{LF}}(t)$, which encodes basic facial features, skin color and lighting, and $\hat{\omega}^i(t)$ – a mapping between coordinate spaces of the texture and the output image. Then, the high-frequency layer of the output image is obtained by warping the predicted texture: $\hat{\mathbf{x}}^i_{\text{HF}}(t) = \hat{\omega}^i(t) \circ \hat{\mathbf{X}}^i(s)$, and is added to a low-frequency component to produce the final image:

$$\hat{\mathbf{x}}^i(t) = \hat{\mathbf{x}}^i_{\text{LF}}(t) + \hat{\mathbf{x}}^i_{\text{HF}}(t)\,. \tag{5.1}$$

- Finally, the *discriminator* network $D\big(\mathbf{x}^i(t), \mathbf{y}^i(t)\big)$, which is a conditional Mehdi Mirza [2014] relativistic Jolicoeur-Martineau [2019] PatchGAN Isola et al. [2017], maps a real or a synthesised target image, concatenated with the target landmark image, into realism scores $\mathbf{s}^i(t)$.

During training, we first input a source image $\mathbf{x}^i(s)$ and a source pose $\mathbf{y}^i(s)$, encoded as a landmark image, into the embedder. The outputs of the embedder are K tensors $\hat{\mathbf{e}}^i_k(s)$, which are used to predict the adaptive parameters of the texture generator and the inference generator. A high-frequency texture $\hat{\mathbf{X}}^i(s)$ of the source image is then synthesized by the texture generator. Next, we input corresponding target keypoints $\mathbf{y}^i(t)$ into the inference generator, which predicts a low-frequency component of the output image $\hat{\mathbf{x}}^i_{\text{LF}}(t)$ directly and a high-frequency component $\hat{\mathbf{x}}^i_{\text{HF}}(t)$ by warping the texture with a predicted field $\hat{\omega}^i(t)$. Finally, the output image $\hat{\mathbf{x}}^i(t)$ is obtained as a sum of these two components.

It is important to note that while the texture generator is manually forced to generate only a high-frequency component of the image via the design of the loss functions, which is described in the next section, we do not specifically constrain it to perform texture inpainting for occluded head parts. This behavior is emergent from the fact that we use two different images with different poses for initialization and loss calculation.

5.3.2 Training process

We use multiple loss functions for training. The main loss function responsible for the realism of the outputs is trained in an adversarial way Goodfellow et al. [2014a]. We also use pixelwise loss to preserve source lightning conditions and perceptual Johnson et al. [2016a] loss to match the source identity in the outputs. Finally, a regularization of the texture mapping adds robustness to the random initialization of the model.

Pixelwise and perceptual losses ensure that the predicted images match the ground truth, and are respectively applied to low- and high-frequency components of the output images. Since usage of pixelwise losses assumes independence of all pixels in the image, the optimization process leads to blurry images Isola et al. [2017], which is suitable for the low-frequency component of the output. Thus the pixelwise loss is calculated by simply measuring mean L_1 distance between the target image and the low-frequency component:

$$\mathcal{L}^G_{\text{pix}} = \frac{1}{HW}||\hat{\mathbf{x}}^i_{\text{LF}}(t) - \mathbf{x}^i(t)||_1 \,. \tag{5.2}$$

On the contrary, the optimization of the perceptual loss leads to crisper and more realistic images Johnson et al. [2016a], which we utilize to train the high-frequency component. To calculate the perceptual loss, we use the stop-gradient operator SG, which allows us to prevent the gradient flow into a low-frequency component. The input generated image is, therefore, calculated as following:

$$\tilde{\mathbf{x}}^i(t) = \text{SG}\big(\hat{\mathbf{x}}^i_{\text{LF}}(t)\big) + \hat{\mathbf{x}}^i_{\text{HF}}(t) \,. \tag{5.3}$$

Following Ha et al. [2019] and Zakharov et al. [2019], our variant of the perceptual loss consists of two components: features evaluated using an ILSVRC (ImageNet) pre-trained VGG19 network Simonyan and Zisserman [2014], and the VGGFace network Parkhi et al. [2015a], trained for face recognition. If we denote the intermediate features of these networks as $\mathbf{f}^i_{k,\text{IN}}(t)$ and $\mathbf{f}^i_{k,\text{face}}(t)$, and their spatial size as $H_k \times W_k$, the objectives can be written as follows:

$$\mathcal{L}^G_{\text{IN}} = \frac{1}{K}\sum_k \frac{1}{H_k W_k}||\tilde{\mathbf{f}}^i_{k,\text{IN}}(t) - \mathbf{f}^i_{k,\text{IN}}(t)||_1 \,, \tag{5.4}$$

$$\mathcal{L}^G_{\text{face}} = \frac{1}{K}\sum_k \frac{1}{H_k W_k}||\tilde{\mathbf{f}}^i_{k,\text{face}}(t) - \mathbf{f}^i_{k,\text{face}}(t)||_1 \,. \tag{5.5}$$

Texture mapping regularization is proposed to improve the stability of the training. In our model, the coordinate space of the texture is learned implicitly, and there are two degrees of freedom that can mutually compensate each other: the position of the face in the texture, and the predicted warping. If, after initial iterations, the major part of the texture is left unused by the model, it can easily compensate that with a more distorted warping field. This artifact of an initialization is not fixed during training, and clearly is not the behavior we need, since we want all the texture to be used to achieve the maximum effective resolution in the outputs. We address the problem by regularizing the warping in the first iterations to be close to an identity mapping:

$$\mathcal{L}^G_{\text{reg}} = \frac{1}{HW}||\omega^i(t) - \mathcal{I}||_1 \,. \tag{5.6}$$

Adversarial loss is optimized by both generators, the embedder and the discriminator networks. Usually, it resembles a binary classification loss function between real and fake images, which discriminator is optimized to minimize, and generators – maximize Goodfellow et al. [2014a]. We

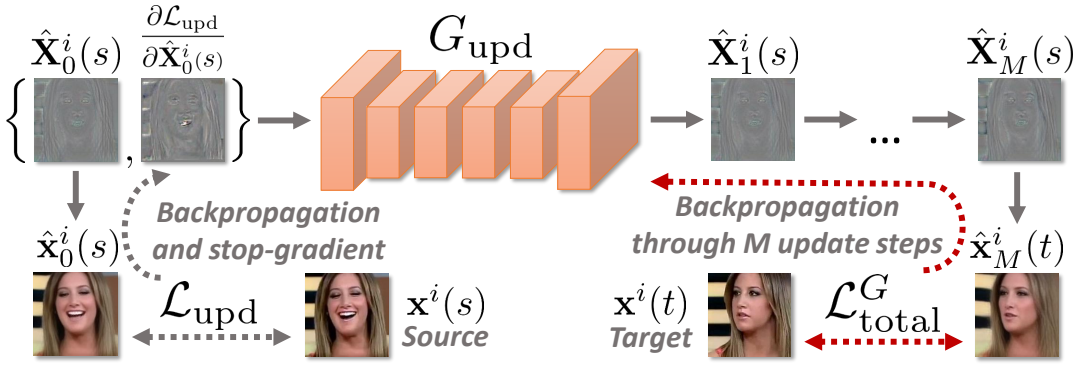


FIGURE 5.3: Texture enhancement network (updater) accepts the current state of the texture and the guiding gradients to produce the next state. The guiding gradients are obtained by reconstructing the source image from the current state of the texture and matching it to the ground-truth via a lightweight updater loss. These gradients are only used as inputs and are detached from the computational graph. This process is repeated M times. The final state of the texture is then used to obtain a target image, which is matched to the ground-truth via the same loss as the one used during training of the main model. The gradients from this loss are then backpropagated through all M copies of the updater network.

follow a large body of previous works Brock et al. [2019b], Ha et al. [2019], Wang et al. [2019], Zakharov et al. [2019] and use a hinge loss as a substitute for the original binary cross entropy loss. We also perform relativistic realism score calculation Jolicoeur-Martineau [2019], following its recent success in tasks such as super-resolution Wang et al. [2018c] and denoising Kim et al. [2019]. Additionally, we use PatchGAN Isola et al. [2017] formulation of the adversarial learning. The discriminator is trained only with respect to its adversarial loss $\mathcal{L}_{\text{adv}}^D$, while the generators and the embedder are trained via the adversarial loss $\mathcal{L}_{\text{adv}}^G$, and also a feature matching loss $\mathcal{L}_{\text{FM}}$ Wang et al. [2018e]. The latter is introduced for better stability of the training.

5.3.3 Texture enhancement

To minimize the identity gap, Zakharov et al. [2019] suggested to fine-tune the generator weights to the few-shot training set. Training on a person-specific source data leads to significant improvement in realism and identity preservation of the synthesized images Zakharov et al. [2019], but is computationally expensive. Moreover, when the source data is scarce, like in one-shot scenario, fine-tuning may lead to over-fitting and performance degradation, which is observed in Zakharov et al. [2019]. We address both of these problems by using a learned gradient descend (LGD) method Andrychowicz et al. [2016] to optimize only the synthesized texture $\hat{\mathbf{X}}^i(s)$. Optimizing with respect to the texture tensor prevents the model from overfitting, while the LGD allows us to perform optimization with respect to computationally expensive objectives by doing forward passes through a pre-trained network.

Specifically, we introduce a lightweight loss function $\mathcal{L}_{\text{upd}}$ (we use a sum of squared errors), that measures the distance between a generated image and a ground-truth in the pixel space, and a

texture updating network G_{upd}, that uses the current state of the texture and the gradient of $\mathcal{L}_{\text{upd}}$ with respect to the texture to produce an update $\Delta\hat{\mathbf{X}}^i(s)$. During fine-tuning we perform M update steps, each time measuring the gradients of $\mathcal{L}_{\text{upd}}$ with respect to an updated texture. The visualization of the process can be seen in Figure 5.3. More formally, each update is computed as:

$$\hat{\mathbf{X}}^i_{m+1}(s) = \hat{\mathbf{X}}^i_m(s) + G_{\text{upd}}\left(\hat{\mathbf{X}}^i_m(s), \frac{\partial \mathcal{L}_{\text{upd}}}{\partial \hat{\mathbf{X}}^i_m(s)}\right), \tag{5.7}$$

where $m \in \{0, \dots, M-1\}$ denotes an iteration number, with $\hat{\mathbf{X}}^i_0(s) \equiv \hat{\mathbf{X}}^i(s)$.

The network G_{upd} is trained by back-propagation through all M steps. For training, we use the same objective $\mathcal{L}^G_{\text{total}}$ that was used during the training of the base model. We evaluate it using a target frame $\mathbf{x}^i(t)$ and a generated frame

$$\hat{\mathbf{x}}^i_M(t) = \hat{\mathbf{x}}^i_{\text{LF}}(t) + \hat{\omega}^i(t) \circ \hat{\mathbf{X}}^i_M(s)\,. \tag{5.8}$$

It is important to highlight that $\mathcal{L}_{\text{upd}}$ is not used for training of G_{upd}, but simply guides the updates to the texture. Also, the gradients with respect to this loss are evaluated using the source image, while the objective in Eq. 5.8 is calculated using the target image, which implies that the network has to produce updates for the whole texture, not just a region "visible" on the source image. Lastly, while we do not propagate any gradients into the generator part of the base model, we keep training the discriminator using the same objective $\mathcal{L}^D_{\text{adv}}$. Even though training the updater network jointly with the base generator is possible, and can lead to better quality (following the success of model agnostic meta-learning Finn et al. [2017] method), we resort to two-stage training due to memory constraints.

5.3.4 Segmentation

The presence of static background leads to a certain degradation of our model for two reasons. Firstly, part of the capacity of the texture and the inference generators has to be spent on modeling high variety of background patterns. Secondly, and more importantly, the static nature of backgrounds in most training videos biases the warping towards identity mapping. We therefore, have found it advantageous to include background segmentation into our model.

We use a state-of-the-art face and body segmentation model Gong et al. [2019b] to obtain the ground truth masks. Then, we add the mask prediction output $\hat{\mathbf{m}}^i(t)$ to our inference generator alongside with its other outputs, and train it via a binary cross-entropy loss $\mathcal{L}_{\text{seg}}$ to match the ground truth mask $\mathbf{m}^i(t)$. To filter out the training signal, related to the background, we have

explored multiple options. Simple masking of the gradients that are fed into the generator leads to severe overfitting of the discriminator. We also could not simply apply the ground truth masks to all the images in the dataset, since the model Gong et al. [2019b] works so well that it produces a sharp border between the foreground and the background, leading to border artifacts that emerge after adversarial training.

Instead, we have found out that masking the ground truth images that are fed to the discriminator with the predicted masks $\hat{\mathbf{m}}^i(t)$ works well. Indeed, these masks are smooth and prevent the discriminator from overfitting to the lack of background, or sharpness of the border. We do not backpropagate the signal from the discriminator and from perceptual losses to the generator via the mask pathway (i.e. we use stop gradient/detach operator $\text{SG}\big(\hat{\mathbf{m}}^i(t)\big)$ before applying the mask). The stop-gradient operator also ensures that the training does not converge to a degenerate state (empty foreground).

5.3.5 Implementation details

All our networks consist of pre-activation residual blocks He et al. [2016b] with LeakyReLU activations. We set a minimum number of features in these blocks to 64, and a maximum to 512. By default, we use half the number of features in the inference generator, but we also evaluate our model with full- and quater-capacity inference part, with the results provided in the experiments section.

We use batch normalization Ioffe and Szegedy [2015] in all the networks except for the embedder and the texture updater. Inside the texture generator, we pair batch normalization with adaptive SPADE layers Wang et al. [2019]. We modify these layers to predict pixelwise scale and bias coefficients using feature maps, which are treated as model parameters, instead of being input from a different network. This allows us to save memory by removing additional networks and intermediate feature maps from the optimization process, and increase the batch size. Also, following Wang et al. [2019], we predict weights for all 1×1 convolutions in the network from the embeddings $\{\hat{\mathbf{e}}^i_k(s)\}$, which includes the scale and bias mappings in AdaSPADE layers, and skip connections in the residual upsampling blocks. In the inference generator, we use standard adaptive batch normalization layers Brock et al. [2019b], but also predict weights for the skip connections from the embeddings.

We do simultaneous gradient descend on parameters of the generator networks and the discriminator using Adam Kingma and Ba [2014] with a learning rate of $2 \cdot 10^{-4}$. We use 0.5 weight for adversarial losses, and 10 for all other losses, except for the VGGFace perceptual loss (Eq. 5.5), which is set to 0.01. The weight of the regularizer (Eq. 5.6) is then multiplicatively reduced by 0.9 every 50 iterations. We train our models on 8 NVIDIA P40 GPUs with the batch size of 48 for the base model, and a batch size of 32 for the updater model. We set unrolling depth M of the

updater to 4 and use a sum of squared errors as the lightweight objective. Batch normalization statistics are synchronized across all GPUs during training. During inference they are replaced with "standing" statistics, similar to Brock et al. [2019b], which significantly improves the quality of the outputs, compared to the usage of running statistics. Spectral normalization is also applied in all linear and convolutional layers of all networks.

Please refer to the supplementary material for a detailed description of our model's architecture, as well as the discussion of training and architectural features that we have adopted.

5.3.6 Experiments

We perform evaluation in multiple scenarios. First, we use the original VoxCeleb2 Chung et al. [2018a] dataset to compare with state-of-the-art systems. To do that, we annotated this dataset using an off-the-shelf facial landmarks detector Bulat and Tzimiropoulos [2017a]. Overall, the dataset contains 140697 videos of 5994 different people. We also use a high-quality version of the same dataset, additionally annotated with the segmentation masks (which were obtained using a model Gong et al. [2019b]), to measure how the performance of our model scales with a dataset of a significantly higher quality. We obtained this version by downloading the original videos via the links provided in the VoxCeleb2 dataset, and filtering out the ones with low resolution. This dataset is, therefore, significantly smaller and contains only 14859 videos of 4242 people, with each video having at most 250 frames (first 10 seconds). Lastly, we do ablation studies on both VoxCeleb2 and VoxCeleb2-HQ, and report on a smartphone-based implementation of the method. For comparisons and ablation studies we show the results qualitatively and also evaluate the following metrics:

- Learned perceptual image patch similarity Zhang et al. [2019] (LPIPS), which measures overall predicted image similarity to ground truth.
- Cosine similarity between the embedding vectors of a state-of-the-art face recognition network Deng et al. [2019] (CSIM), calculated using the synthesized and the target images. This metric evaluates the identity mismatch.
- Normalized mean error of the head pose in the synthesized image (NME). We use the same network Bulat and Tzimiropoulos [2017a], which was used for the annotation of the dataset, to evaluate the pose of the synthesized image. We normalize the error, which is a mean euclidean distance between the predicted and the target points, by the distance between the eyes in the target pose, multiplied by 10.
- Multiply-accumulate operations (MACs), which measure the complexity of each method. We exclude from the evaluation initialization steps, which are calculated only once per avatar.

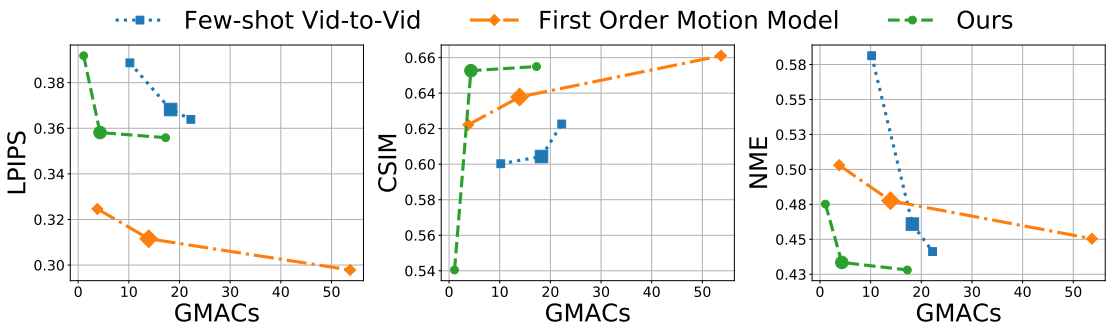


FIGURE 5.4: In order to evaluate a quality against performance trade off, we train a family of models with varying complexity for each of the compared methods. For quality metrics, we have compared synthesized images to their targets using a perceptual image similarity (LPIPS ↓), identity preservation metric (CSIM ↑), and a normalized pose error (NME ↓). We highlight a model which was used for the comparison in Figure 5.5 with a bold marker. We observe that our model outperforms the competitors in terms of identity preservation (CSIM) and pose matching (NME) in the settings, when models' complexities are comparable. In order to better compare with FOMM, we did a user study, where users have preferred the image generated by our model to FOMM 59.6% of the time.

The test set in both datasets does not intersect with the train set in terms of videos or identities. For evaluation, we use a subset of 50 test videos with different identities (for VoxCeleb2, it is the same as in Zakharov et al. [2019]). The first frame in each sequence is used as a source. Target frames are taken sequentially at 1 FPS.

We only discuss most important results in the main paper. For additional qualitative results and comparisons please refer to the supplementary materials.

5.3.6.1 Comparison with the state-of-the-art methods

We compare against three state-of-the-art systems: Few-shot Talking Heads Zakharov et al. [2019], Few-shot Vid-to-Vid Wang et al. [2019] and First Order Motion Model Siarohin et al. [2019c]. The first system is a problem-specific model designed for avatar creation. Few-shot Vid-to-Vid is a state-of-the-art video-to-video translation system, which has also been successfully applied to this problem. First Order Motion Model (FOMM) is a general motion transfer system that does not use precomputed keypoints, but can also be used as an avatar. We believe that these models are representative of the most recent and successful approaches to one-shot avatar generation. We also acknowledge the work of Ha et al. [2019], but do not compare to them extensively due to unavailability of the source code, pretrained models or pre-calculated results. A small-scale qualitative comparison is provided in the supplementary materials. Additionally, their method is limited to the usage of 3D keypoints, while our method does not have such restriction. Lastly, since Few-shot Vid-to-Vid is an autoregressive model, we use a full test video sequence for evaluation (25 FPS) and save the predicted frames at 1 FPS.

FIGURE 5.5: Comparison on a VoxCeleb2 dataset. The task is to reenact a **target** image, given a **source** image and target keypoints. The compared methods are Few-shot Talking Heads Zakharov et al. [2019], Few-shot Vid-to-Vid Wang et al. [2019], First Order Motion Model (FOMM) Siarohin et al. [2019c] and our proposed Bi-layer Model. For each method, we used the models with a similar number of parameters, and picked source and target images to have diverse poses and expressions, in order to highlight the differences between the compared methods.

Importantly, the base models in these approaches have a lot of computational complexity, so for each method we evaluate a family of models by varying the number of parameters. The performance comparison for each family is reported in Figure 5.4 (with Few-shot Talking Heads being excluded from this evaluation, since their performance is much worse than the compared methods). Overall, we can see that our model's family outperforms competing methods in terms of pose error and identity preservation, while being, on average, up to an order of magnitude faster. To better compare with FOMM in terms of image similarity, we have performed a user study, where we asked crowd-sourced users which generated image better matches the ground truth. In total, 361 users evaluated 1600 test pairs of images, with each one seeing on average 21 pairs. In 59.6% of comparisons, the result of our medium model was preferred to a medium sized model of FOMM.

Another important note is on how the complexity was evaluated. In Few-shot Vid-to-Vid we have additionally excluded from the evaluation parts that are responsible for the temporal consistency, since other compared methods are evaluated frame-by-frame and do not have such overhead.

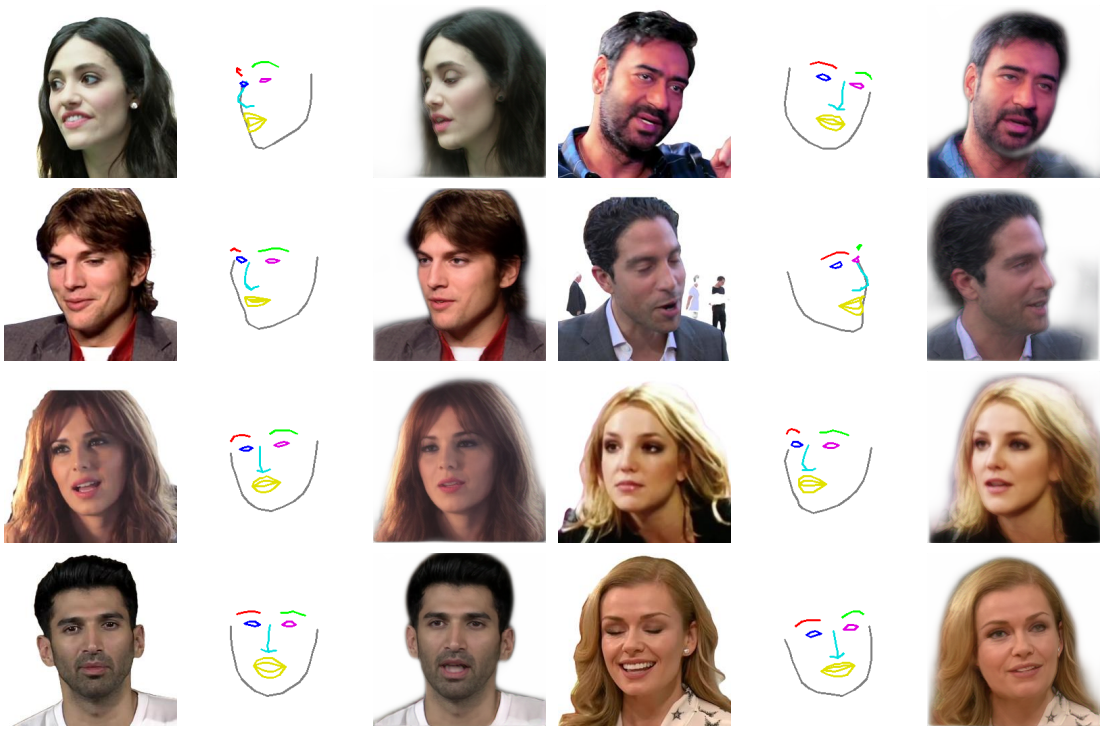


FIGURE 5.6: High quality synthesis results. We can see that our model is both capable of viewpoint extrapolation and low identity gap synthesis. The architecture in this experiment has the same number of parameters as the medium architecture in the previous comparison.

Also, in FOMM we have excluded the keypoints extractor network, because this overhead is shared implicitly by all the methods via usage of the precomputed keypoints.

We visualize the results for medium-sized models of each of the compared methods in Figure 5.5. Since all methods perform similarly in case when source and target images have marginal differences, we have shown the results where a source and a target have different head poses. In this extrapolation setting, our method has a clear advantage, while other methods either introduce more artifacts or more blurriness.

5.3.6.2 Evaluation on high-quality images.

Next, we evaluate our method on the high-quality dataset and present the results in Figure 5.6. Overall, in this case, our method is able to achieve a smaller identity gap, compared to the dataset with the background. We also show the decomposition between the texture and a low frequency component in Figure 5.7. Lastly, in Figure 5.8, we show that our texture enhancement pipeline allows us to render small person-specific features like wrinkles and moles on out-of-domain examples. For more qualitative examples, as well as reenactment examples with a driver of a different person, please refer to the supplementary materials.

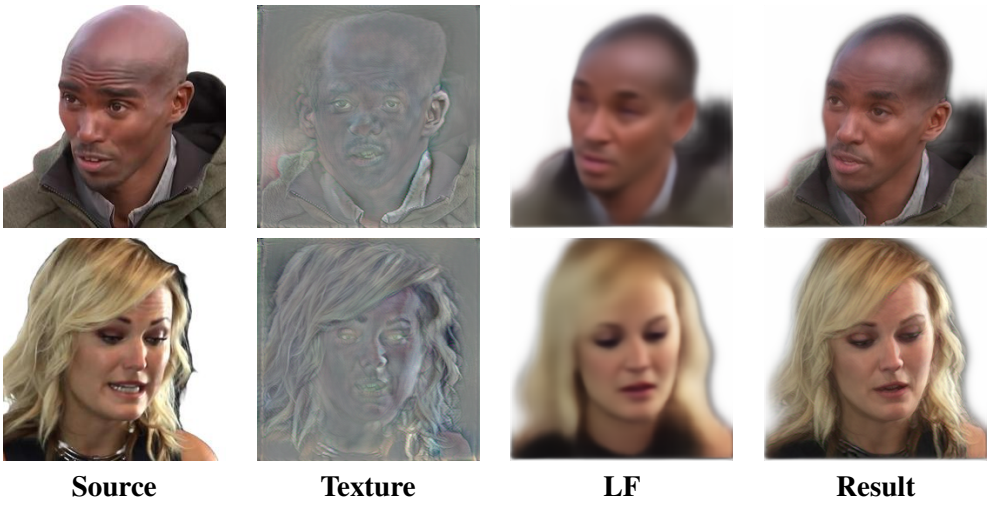


FIGURE 5.7: Detailed results on the generation process of the output image. **LF** denotes a low-frequency component.

FIGURE 5.8: Our method can preserve a lot of details in the facial features, like the famous Marylin's mole.

5.3.6.3 Smartphone-based implementation.

We train our model using PyTorch PyT and then port it to smartphones with Qualcomm Snapdragon chips. There are several frameworks which provide APIs for mobile inference on such devices. From our experiments, we measured the Snapdragon Neural Processing Engine (SNPE) SNP to be about 1.5 times faster than PyTorch Mobile PyT and up to two times faster than TensorFlow Lite TFL. The medium-sized model ported to the Snapdragon 855 (Adreno 640 GPU, FP16 mode) takes 42 ms per frame, which is sufficient for real-time performance, given that the keypoint tracking is being run in parallel, e.g. on a mobile CPU.

Method	LPIPS ↓	CSIM ↑	NME ↓
VoxCeleb2			
Baseline	0.377	0.547	0.447
Ours	0.370	0.595	0.441
+Updater	**0.358**	**0.653**	**0.433**
VoxCeleb2-HQ			
Ours	0.313	0.432	0.476
+Updater	**0.298**	**0.649**	**0.456**

TABLE 5.1: Ablation studies of our approach. We first evaluate the baseline method without AdaSPADE or adaptive skip connections. Then we add these layers, following Wang et al. [2019], and observe significant quality improvement. Finally, our updater network provides even more improvement across all metrics, especially noticeable in the high-quality scenario.

5.3.6.4 Ablation study.

Finally, we evaluate the contribution of individual components. First, we evaluate the contribution of adaptive SPADE layers in the texture generator (by replacing them with adaptive batch normalization and per-pixel biases) and adaptive skip-connections in both generators. A model with these features removed makes up our baseline. Lastly, we evaluate the contribution of the updater network. The results can be seen in Table 5.1 and Figure 5.9. We evaluate the baseline approach only on a VoxCeleb2 dataset, while the full models with and without the updater network are evaluated on both low- and high-quality datasets. Overall, we see a significant contribution of each component with respect to all metrics, which is particularly noticeable in the high-quality scenario. In all ablation comparisons, medium-sized models were used.

5.4 Conclusion

We have proposed a new neural rendering-based system that creates head avatars from a single photograph. Our approach models person appearance by decomposing it into two layers. The first layer is a pose-dependent coarse image that is synthesized by a small neural network. The second layer is defined by a pose-independent texture image that contains high-frequency details and is generated offline. During test-time it is warped and added to the coarse image to ensure high effective resolution of synthesized head views. We compare our system to analogous state-of-the-art systems in terms of visual quality and speed. The experiments show up to an order of magnitude inference speedup over previous neural head avatar models, while achieving state-of-the-art quality. We also report on a real-time smartphone-based implementation of our system.

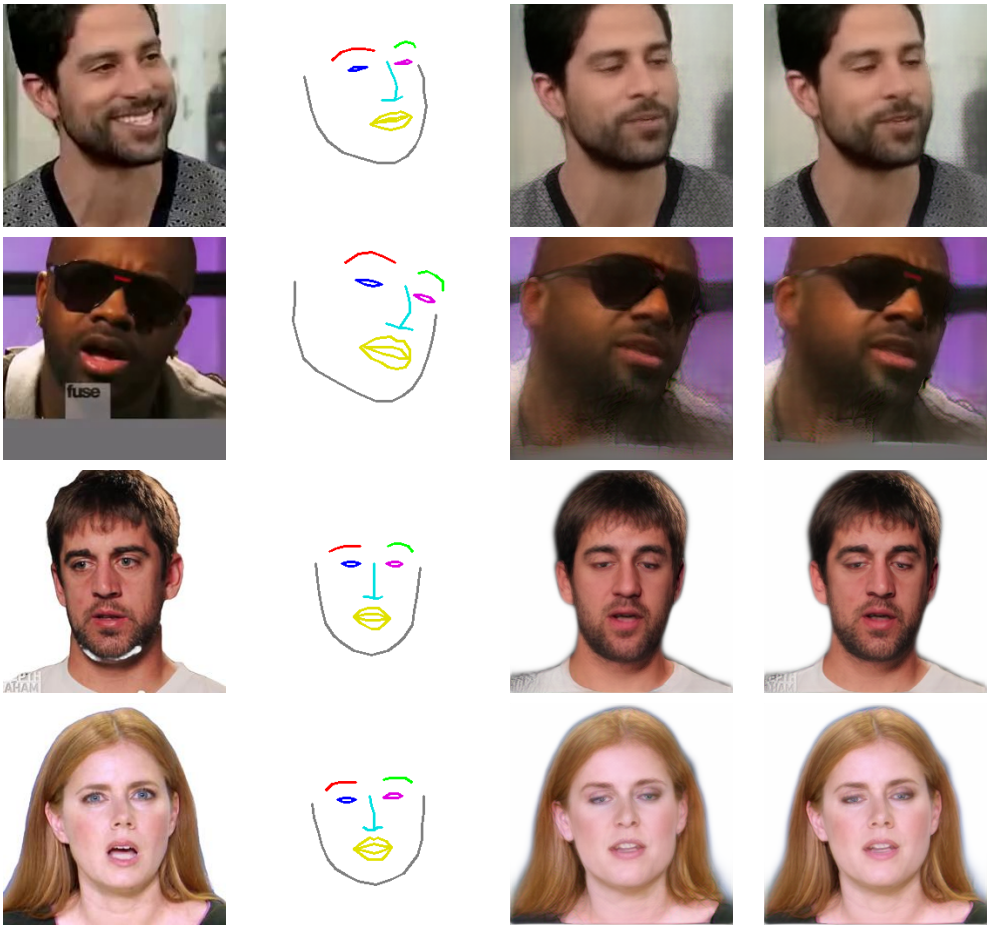


FIGURE 5.9: Examples from the ablation study on VoxCeleb2 (first two rows) and VoxCeleb2-HQ (last two rows).

Chapter 6

MegaPortraits: One-shot Megapixel Neural Head Avatars

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 show 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.

This work was published as: Nikita Drobyshev, Evgeny Chelishev, Aleksei Ivakhnenko, Taras Khakhulin, Victor Lempitsky, and Egor Zakharov. *MegaPortraits: One-shot Megapixel Neural Head Avatars*. ACM Conference on Multimedia (ACMMM), 2022.

Supplementary materials are hosted on the project page: https://samsunglabs.github.io/MegaPortraits

6.1 Introduction

Neural head avatars Burkov et al. [2020], Doukas et al. [2021b], Ha et al. [2020], Kim et al. [2018b], Lombardi et al. [2018b, 2019], Park et al. [2021a,b], Siarohin et al. [2019a,b], Thies et al. [2019b], Wang et al. [2021], Zakharov et al. [2019, 2020] offer a new fascinating way to create 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 Doukas et al. [2021b], Siarohin et al. [2019b], Wang et al. [2021], Zakharov et al. [2020]. Such methods leverage extensive pre-training on the large datasets of videos of different people Chung et al. [2018b], Wang et al. [2021] to create their avatars in the one-shot mode using the 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 is not trivial to bypass by collecting a higher resolution dataset. This is because an appropriate dataset needs to be large-scale and diverse at the same time. It needs to 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 Chung et al. [2018b], Wang et al. [2021] that meet these criteria are limited in resolution. As a result, even the most recent one-shot avatar systems Wang et al. [2021] 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 Wang et al. [2021] and propose a new way to combine it with the latent motion representations Burkov et al. [2020]. We also propose to use a novel contrastive loss 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 Yang et al. [2020] for the task of cross-reenactment. We are thus the first to demonstrate neural head avatars in proper megapixel resolution.

Lastly, since a lot of practical applications of neural avatars require real-time or faster than real-time rendering, we distill our megapixel model into ten times faster student model that runs at 70 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). 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.

6.2 Related work

6.2.1 Implicit functions for radiance modeling

The recent success of neural implicit scene representations Mildenhall et al. [2020] for the problem of 3D reconstruction has inspired several works on the so-called 4D head avatars Gafni et al. [2021], Lombardi et al. [2018b, 2019], Park et al. [2021a,b], Yang et al. [2021], 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 Park et al. [2021a,b], Yang et al. [2021], use pre-trained motion extractors Gafni et al. [2021] or pre-computed coarse meshes Lombardi et al. [2018b, 2019]. 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.

6.2.2 Direct generation of images via convolutional networks

Direct generation of videos via convolutional neural networks, conditioned on both 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 Kim et al. [2018b], Thies et al. [2019b], the follow-up works added few-shot and one-shot capabilities Burkov et al. [2020], Doukas et al. [2021b], Siarohin et al. [2019a,b], Wang et al. [2021], Zakharov et al. [2019, 2020]. Most of these works use explicit representations for the motion, such as keypoints or blendshapes, while others Burkov et al. [2020] 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.

6.2.3 Single image super-resolution

The resolution of the talking head models is currently upper bounded by the available video datasets Chung et al. [2018b], Wang et al. [2021], which contain videos of at most 512×512 resolution. This problem restricts the enhancement of the output quality further on the existing datasets using the standard high-quality image and video synthesis techniques Wang et al. [2018d,g]. 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 Yang et al. [2020]. 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.

6.3 Method

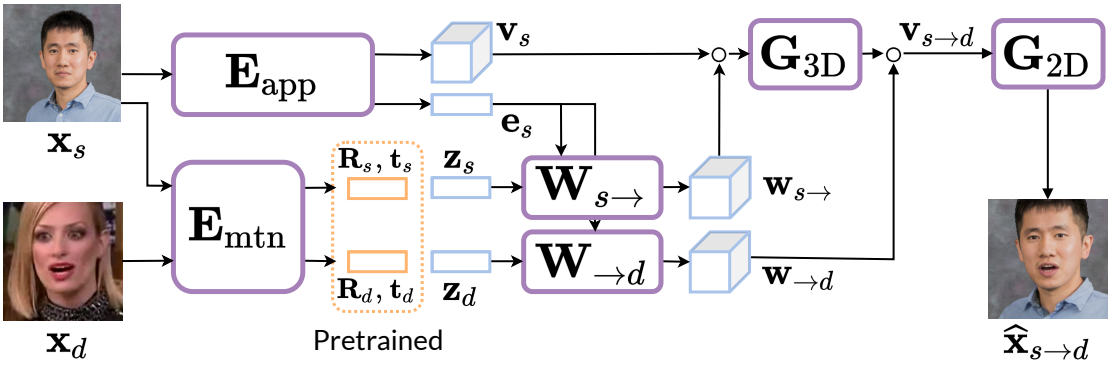


FIGURE 6.1: Overview of our base model. To encode the appearance of the source frame, we predict volumetric features $\mathbf{v}_s$ and a global descriptor $\hat{\mathbf{e}}_s$ from the source image via an appearance encoder $\mathbf{E}_{\text{app}}$. In parallel, we predict the motion representations 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 $\omega_{s\rightarrow}$ and $\omega_{\rightarrow d}$ via the separate warping generators $\mathbf{W}_{s\rightarrow}$ and $\mathbf{W}_{\rightarrow d}$. 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 $\mathbf{v}_{s\rightarrow d}$ is orthographically projected into 2D features and processed by a 2D convolutional network $\mathbf{G}_{\text{2D}}$, which predicts an output image $\hat{\mathbf{x}}_{s\rightarrow d}$.

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.

6.3.1 Base model

During the first stage, we train our base model (Figure 6.1) 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 $\hat{\mathbf{e}}_s$. 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 $(\mathbf{R}_s, \mathbf{t}_s, \mathbf{z}_s, \hat{\mathbf{e}}_s)$ is then input into a warping generator $\mathbf{W}_{s\to}$ to produce a 3D warping field $\omega_{s\to}$, 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 $(\mathbf{R}_d, \mathbf{t}_d, \mathbf{z}_d, \hat{\mathbf{e}}_s)$ is fed into a separate warping generator $\mathbf{W}_{\to d}$, which output $\omega_{\to d}$ is used to impose the driver motion. The final 4D volumetric features are therefore obtained in the following way:

$$\mathbf{v}_{s\to d} = \omega_{\to d} \circ \mathbf{G}_{\text{3D}}(\omega_{s\to} \circ \mathbf{v}_s), \tag{6.1}$$

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 Wang et al. [2021]. However, a significant difference with Wang et al. [2021] is that we do not use keypoints to represent expression and instead rely on the latent descriptor Burkov et al. [2020], 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 Burkov et al. [2020] 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 Wang et al. [2021]. We denote this operation as $\mathcal{P}$. The resulting 2D feature map is decoded into the output image by a 2D convolutional network $\mathbf{G}_{\text{2D}}$:

$$\hat{\mathbf{x}}_{s\to d} = \mathbf{G}_{\text{2D}}\big(\mathcal{P}(\mathbf{v}_{s\to d})\big). \tag{6.2}$$

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

$$\hat{\mathbf{x}}_{s\to d} = \mathbf{G}_{\text{base}}(\mathbf{x}_s, \mathbf{x}_d). \tag{6.3}$$

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 Johnson et al. [2016b] and GAN Wang et al. [2018g] 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 Zhu et al. [2017b] 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) Deng et al. [2009] pre-trained VGG19 Simonyan and Zisserman [2015b] to match the general content of the images, VGGFace Parkhi et al. [2015b] 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 Fischer et al. [2018]. For more details on the training and usage of the 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:

$$\mathcal{L}_{\text{per}} = w_{\text{IN}}\mathcal{L}_{\text{IN}} + w_{\text{face}}\mathcal{L}_{\text{face}} + w_{\text{gaze}}\mathcal{L}_{\text{gaze}}. \tag{6.4}$$

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 Zhu et al. [2017b] with a hinge adversarial loss alongside the generator $\mathbf{G}_{\text{base}}$. We also include a standard feature-matching loss Wang et al. [2018g] to improve the training

stability. The GAN loss for the generator can therefore be expressed as follows:

$$\mathcal{L}_{\text{GAN}} = w_{\text{adv}}\mathcal{L}_{\text{adv}} + w_{\text{FM}}\mathcal{L}_{\text{FM}}. \tag{6.5}$$

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: $\mathcal{P} = \big\{(\mathbf{z}_{s\to d}, \mathbf{z}_d), (\mathbf{z}_{s^*\to d}, \mathbf{z}_d)\big\}$, and the *negative pairs*: $\mathcal{N} = \big\{(\mathbf{z}_{s\to d}, \mathbf{z}_{d^*}), (\mathbf{z}_{s^*\to d}, \mathbf{z}_{d^*})\big\}$. These pairs are used to calculate the following cosine distance:

$$d(\mathbf{z}_i, \mathbf{z}_j) = s \cdot \big(\langle \mathbf{z}_i, \mathbf{z}_j \rangle - m\big), \tag{6.6}$$

where both s and m are hyperparameters. This distance is then used to calculate a large margin cosine loss (CosFace) Wang et al. [2018a]:

$$\mathcal{L}_{\text{cos}} = -\sum_{(\mathbf{z}_k, \mathbf{z}_l)\in\mathcal{P}} \log \frac{\exp\big\{d(\mathbf{z}_k, \mathbf{z}_l)\big\}}{\exp\big\{d(\mathbf{z}_k, \mathbf{z}_l)\big\} + \sum\limits_{(\mathbf{z}_i, \mathbf{z}_j)\in\mathcal{N}} \exp\big\{d(\mathbf{z}_i, \mathbf{z}_j)\big\}}. \tag{6.7}$$

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

$$\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{per}} + \mathcal{L}_{\text{GAN}} + w_{\text{cos}}\mathcal{L}_{\text{cos}}. \tag{6.8}$$

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 Suvorov et al. [2021] and use the following systems for matting Gong et al. [2019a], Ke et al. [2022]. The background is combined with the predicted image via alpha-compositing using a calculated matte. For more details, please refer to the supplementary materials.

6.3.2 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 Karras et al. [2019] 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:

$$\mathcal{L}_{\text{cyc}}^{\text{c}} = \mathcal{L}_{\text{MAE}}\big(\text{DS}_4(\hat{\mathbf{x}}_{\text{c}}), \text{DS}_8(\hat{\mathbf{x}}_{\text{c}}^{\text{HR}})\big), \tag{6.9}$$

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 $\hat{\mathbf{x}}_{c}^{\text{HR}}$ and $\mathbf{x}^{\text{HR}}$, and the cycle-consistency loss $\mathcal{L}_{\text{cyc}}^{\text{c}}$:

$$\mathcal{L}_{\text{enh}} = \mathcal{L}_{\text{GAN}} + w_{\text{MAE}}\mathcal{L}_{\text{MAE}} + w_{\text{adv}}^{\text{c}}\mathcal{L}_{\text{adv}}^{\text{c}} + w_{\text{cyc}}^{\text{c}}\mathcal{L}_{\text{cyc}}^{\text{c}}. \tag{6.10}$$

6.3.3 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

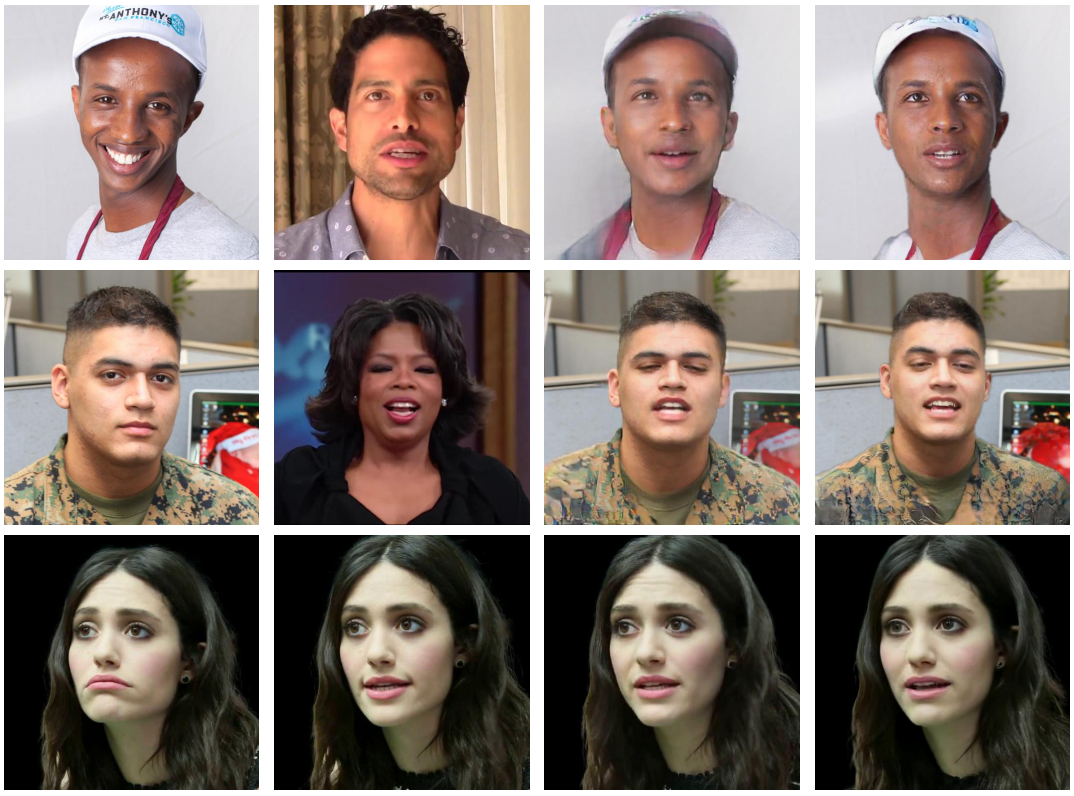


FIGURE 6.2: A qualitative comparison of head avatar systems in cross-reenactment scenario (top two rows) and self-reenactment scenario (bottom row) at 512px resolution. In cross-reenactment, we can see that our approach achieves better preservation of motion and appearance than the previous state-of-the-art (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.

following two images:

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

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

6.4 Experiments

We use multiple datasets to train and evaluate our model: VoxCeleb2 Chung et al. [2018b] and VoxCeleb2HQ video datasets, and FFHQ Karras et al. [2019] 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 Su et al. [2020c]. 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

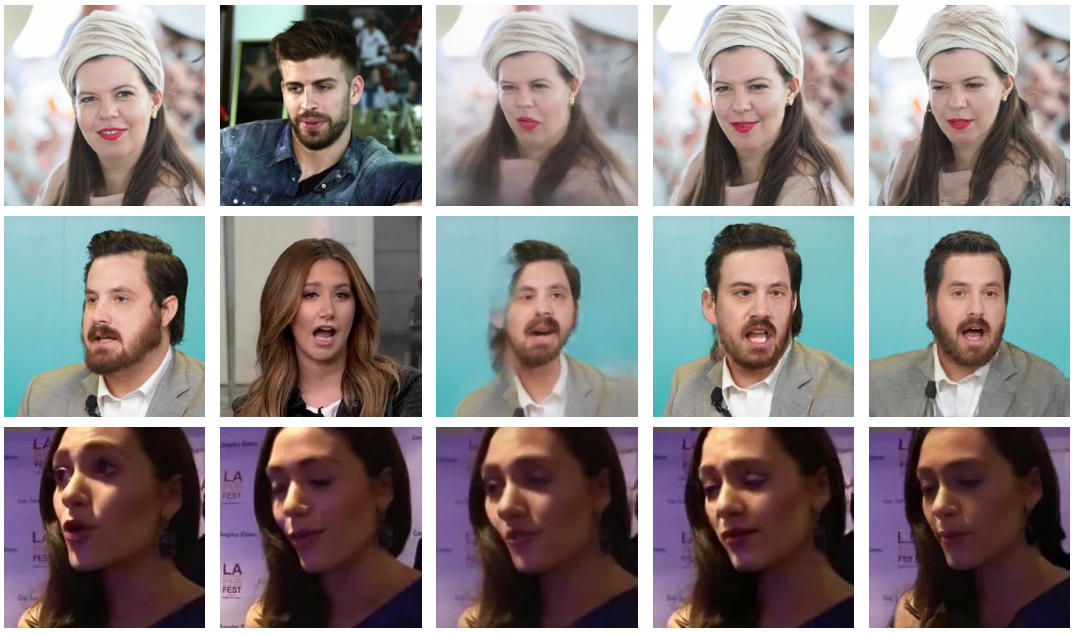


FIGURE 6.3: A qualitative comparison of head avatar systems in cross-reenactment scenario (top two rows) and self-reenactment scenario (bottom row) at 256×256 resolution. Our system significantly outperforms the competitors in cross-reenactment, achieving more faithful motion and appearance preservation 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.

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.

6.4.1 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 Loshchilov and Hutter [2019] 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.

6.4.2 Baseline methods

We compare our base model with the following systems.

Face Vid-to-vid (Face-V2V) Wang et al. [2021] 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) Siarohin et al. [2019b] 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 Doukas et al. [2021b] system, in which the expression coefficients of the 3D morphable model Blanz and Vetter [1999] are used as a motion representation. These coefficients are calculated using a pre-trained dense 3D keypoints regressor Deng et al. [2020]. 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).

6.4.3 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 Wang et al. [2021] 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) Heusel et al. [2017b] 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) Zakharov et al. [2019] 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

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 6.1: 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.

percentage of examples where each method was picked. We have conducted our experiment on approximately 2,000 crowd-sourced people, and each evaluation sample was shown, on average, to twenty different users.

The qualitative results are shown in Figures 6.2-6.3, and the quantitative metrics are presented in Table 6.1. Overall, we can see that our method outperforms all competitors by some margin. Furthermore, the first two rows in Figure 6.3 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 6.2, 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 6.1), including both user studies.

FIGURE 6.4: 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.

6.4.4 Self-reenactment evaluation

For the self-reenactment experiments, we use the same pre-trained models as for the cross reenactment and evaluate them on the samples from the VoxCeleb2 and VoxCeleb2HQ evaluation sets. 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) Wang et al. [2004b], and the Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. [2018a].

We notice that qualitatively we achieve similar performance to the competitors, especially in the face and hair regions (papers/megaportraits/figures 6.2-6.3, 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 6.1) 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.

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 6.2: 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.

6.4.5 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 augmentation transforms 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 a state-of-the-art face super-resolution system (HiFaceGAN) Yang et al. [2020]. The results are presented in Figure 6.4, and Table 6.2. In the quantitative comparison, we use an additional image quality assessment metric (IQA) Su et al. [2020c] 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 6.5 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.

6.4.6 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 6.6.

FIGURE 6.5: 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.

FIGURE 6.6: 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.

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.

6.5 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. The use of the student network also increase the rendering speed while achieving similar quality of renders to our full one-shot model.

FIGURE 6.7: 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).

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 6.7). 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 architecture choices. Lastly, our system lacks the modeling of shoulders motion. Addressing the above-mentioned issues remains our future work.

Chapter 7

Realistic One-shot Mesh-based Head Avatars

Abstract

We present a system for the creation of realistic one-shot mesh-based (ROME) human head avatars. From a single photograph, our system estimates the head mesh (with person-specific details in both the facial and non-facial head parts) as well as the neural texture encoding, local photometric and geometric details. The resulting avatars are rigged and can be rendered using a deep rendering network, which is trained alongside the mesh and texture estimators on a dataset of in-the-wild videos. In the experiments, we observe that our system performs competitively both in terms of head geometry recovery and the quality of renders, especially for cross-person reenactment.

This work was published as: Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. *Realistic One-shot Mesh-based Head Avatars.* European Conference on Computer Vision (ECCV), 2022.

Supplementary materials are hosted on the project page: https://samsunglabs.github.io/rome

7.1 Introduction

Personalized human avatars are becoming the key technology across several application domains, such as telepresence, virtual worlds, online commerce. In many cases, it is sufficient to personalize only a part of the avatars' body. The remaining body parts can then be either chosen from a certain library of assets or omitted from the interface. Towards this end, many applications require personalization at the head level, i.e. creating person-specific head models. Creating personalized heads is an important and viable intermediate step between personalizing just face (which is often insufficient) and creating personalized full-body models, which is a much harder task that limits quality of the resulting models and/or requires cumbersome data collection.

Acquiring human avatars from a single photograph ("one-shot") offers the highest convenience for users, yet is particularly challenging and requires strong priors on human geometry and appearance. For faces, parametric models are long known to offer good personalization solutions Blanz and Vetter [1999]. Face models can also be learned from a relatively small dataset of 3D scans, and represent geometry using meshes and appearance using textures, which makes such models compatible with many computer graphics applications and pipelines. On the other hand, parametric face models cannot be trivially expanded to the whole head region due to large geometric variability of the non-facial parts such as hair and neck.

In this work, we extend parametric mesh-based modeling to the human heads. In order to learn the increased geometric and photometric variability (compared to faces), we learn our parametric models directly from a large dataset of in-the-wild videos Chung et al. [2018b]. We use neural networks to parameterize both the geometry and the appearance. For the appearance modeling, we follow the deferred neural rendering Thies et al. [2019a] paradigm and use a combination of neural textures and rendering networks. A neural rendering framework Ravi et al. [2020] is used to enable end-to-end training and to achieve high visual realism of the resulting head models. After training, both the geometric and the appearance neural networks can be conditioned on the information extracted from a single photograph, enabling one-shot realistic avatar generation.

To the best of our knowledge, our system is the first that is capable of creating realistic personalized human head models in a rigged mesh format from a single photograph. This distinguishes our model from a growing class of approaches that recover neural head avatars that lack explicit geometry Siarohin et al. [2019b], Wang et al. [2021], Zakharov et al. [2019, 2020], from another big class of approaches that can personalize the face region but not the whole head Blanz and Vetter [1999], Feng et al. [2020], Kim et al. [2018c], Thies et al. [2016a], and from commercial systems that create non-photorealistic mesh avatars from a single image Ava, Pinscreen. Alongside our full model, we also discuss a simplification of it based on linear blendshape basis and show how such simplification and a corresponding feedforward predictor for blendshape coefficients can be trained (on the same video dataset).

FIGURE 7.1: Results of our system. It creates mesh-based avatars from a single photo (**Source**). The resulting avatars are rigged, i.e., can be driven by the animation parameters from a different frame (**Driver**) and use an underlying mesh that has a predefined topology and reflects person-specific geometry (**Mesh**). The realism of the renders is enhanced by the use of neural rendering, which is trained jointly with the geometric model generation on a dataset of in-the-wild videos.

Overall, our contributions are as follows:

- We present a system that is able to create mesh-based rigged head avatars from a single image.
- We show how such system can be trained using in-the-wild videos by using modern differentiable renderers.
- Our approach achieves state-of-the-art in the task of one-shot head mesh reconstruction.
- We discuss how a simplified linear blendshape model, which approximates our full system, can be trained.

Below, we refer to the avatars generated by our system as ROME avatars (Realistic One-shot Mesh-based avatars).

7.2 Related work

7.2.1 Parametric models of human faces

3D face reconstruction has been actively developed over decades for face tracking and alignment Guo et al. [2020], Hassner et al. [2015], face recognition Blanz et al. [2002], Tran et al. [2017], and generative modelling Kim et al. [2018c], Lombardi et al. [2018b, 2019], Mildenhall et al. [2020], Ramon et al. [2021], Thies et al. [2016a]. In all these scenarios, statistical mesh-based models (aka parametric models) Blanz and Vetter [1999] remain one of the widely used tools Egger et al. [2020], Ploumpis et al. [2021]. Such models impose a strong prior on the space of possible reconstructions. State-of-the-art parametric models for human heads consist of rigged meshes Li et al. [2017] which support a diverse range of animations with rigid motions for

jaw, neck, and eyeballs, as well as via disentangled shape and expression coefficients. However, they only provide reconstructions for face, ears, neck, and forehead regions, which limits the range of applications. The inclusion of full head reconstruction (i.e., hair and clothing) into these parametric models is possible, but in order to do that current approaches require significantly more training data to be gathered in the form of 3D scans. Instead, in our work, we propose to leverage existing large-scale datasets Chung et al. [2018b] of in-the-wild videos via the learning-by-synthesis paradigm without using any additional 3D scans.

7.2.2 Neural 3D human head models

While parametric models provide sufficient reconstruction quality for many downstream applications, they are not able to model very fine details that are needed for photorealistic modeling. In recent years, approaches that model the very complex geometry and/or appearance of humans using high-capacity deep neural networks. Some of these works use strong human-specific priors Feng et al. [2020], Lombardi et al. [2021], Ramon et al. [2021], Saito et al. [2020b]. Others, fit high-capacity networks to data without the use of such priors Kellnhofer et al. [2021], Lombardi et al. [2018b, 2019], Ma et al. [2021], Mildenhall et al. [2020], Oechsle et al. [2021], Park et al. [2019]. The methods in this class differ by the type of data structure used to represent the geometry, namely, mesh-based Feng et al. [2020], Lombardi et al. [2018b, 2019], point-based Ma et al. [2021], Zakharkin et al. [2021], and implicit models Lombardi et al. [2021], Mildenhall et al. [2020], Oechsle et al. [2021], Park et al. [2019], Ramon et al. [2021], Saito et al. [2020b], Yenamandra et al. [2021].

Mesh-based models arguably represent the most convenient class of methods for downstream applications. They provide better rendering quality and better temporal stability than point-based neural rendering. Also, unlike methods based on implicit geometry, mesh-based methods allow to preserve topology and rigging capability, and are also much faster during fitting and/or rendering. However, currently, mesh-based methods either severely limit the range of deformations Feng et al. [2020], making it infeasible to learn complex geometry like hair or clothing, or operate in the multi-shot scenario and require an excessive number of 3D scans as training data Lombardi et al. [2018b, 2019]. Our proposed method is also mesh-based, but we allow the prediction of complex deformations without 3D supervision, lifting the limitations of the previous works.

7.2.3 One-shot neural head models

Advances in neural networks also led to the development of methods that directly predict images using large ConvNets operating in the 2D image domain, with effectively no underlying 3D geometry Siarohin et al. [2019b], Zakharov et al. [2019, 2020] or with very coarse 3D

geometry Wang et al. [2021]. These methods achieve state-of-the-art realism Wang et al. [2021], use in-the-wild images or videos with no 3D annotations for training, and can create avatars from a single image. However, the lack of an explicit geometric model, makes these models incompatible with many real-world applications, and limits the span of camera poses that can be handled by these methods.

7.2.4 Neural mesh rendering

Recently, approaches that combines explicit data structures (point clouds or meshes) with neural image generation have emerged. For mesh-based geometry, this method has been pioneered and popularized by the Deferred Neural Rendering system Thies et al. [2019a]. This class of methods also benefit from the recent advances in differentiable mesh rendering Laine et al. [2020], Liu et al. [2019], Ravi et al. [2020]. Neural mesh rendering uses 2D convolutional networks to model complex photometric properties of surfaces, and achieves high realism of renders with fine details even when such details are missing in the underlying geometric model. In this work, we adapt these advances to human head modeling and combine them with learning from large datasets of in-the-wild videos.

7.3 Method

Our goal is to build a system that jointly learns to produce photorealistic renders of human heads, as well as to estimate their 3D meshes using only a *single image* and without any 3D supervision.

To achieve that, we use a large-scale dataset Chung et al. [2018b] of in-the-wild videos with talking speakers. All frames in each video are assumed to depict the same person in the same environment (defined by lighting, hairstyle, and person's clothing).

At each training step, we sample two random frames $\mathbf{x}_s$ and $\mathbf{x}_t$ from a random training video. Our goal is to reconstruct and to render the target image $\hat{\mathbf{x}}_t$ given a) the personal details and the face shape extracted from the source image $\mathbf{x}_s$, as well as b) the head pose, the facial expression, and the camera pose estimated from the target image $\mathbf{x}_t$. The final reconstruction loss is backpropagated and used to update the parameters of the model components.

After training, we can create a personalized head model by estimating all parameters from a single image. This model can then be *animated* using face tracking parameters extracted from any talking head sequence and rendered from a range of viewpoints similar to those present in the training dataset.

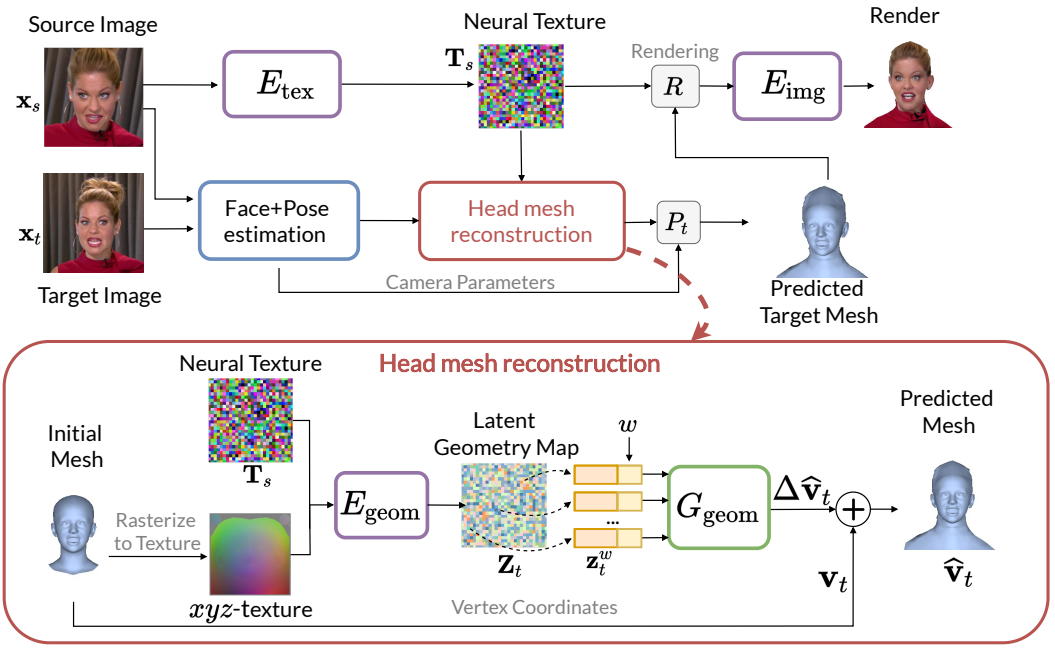


FIGURE 7.2: Overview of our approach and the detailed scheme of the head mesh reconstruction. Given the source photo, we first estimate a *neural texture* that encodes local geometric and photometric details of visible and occluded parts. We then use a pre-trained system Feng et al. [2020] for face reconstruction to estimate an initial mesh with a reconstructed facial part. We call this step face and 3D pose estimation. During head mesh reconstruction (bottom), using the estimated neural texture and the initial mesh, we predict the offsets for the mesh vertices, which do not correspond to a face. The offsets are predicted using a combination of a convolutional network E_{geom} and a perceptron network G_{geom}. We then render the personalized head mesh using the camera parameters, estimated by a pre-trained regressor Feng et al. [2020], while superimposing the predicted neural texture. Finally, the rendering network E_{img} estimates the RGB image and the mask from the render.

7.3.1 Model overview

In our model, we jointly train multiple neural networks that perform rendering and mesh reconstruction. The training pipeline proceeds as follows (Figure 7.2):

Latent texture estimation. The source image $\mathbf{x}_s$ is encoded into a neural texture $\mathbf{T}_s$ that contains local person-specific details (describing both appearance and geometry). The encoding is done by a convolutional neural network E_{tex}.

Face and 3D pose estimation. In parallel, we apply a pre-trained DECA system Feng et al. [2020] for face reconstruction to both the source and the target image. DECA estimates facial shape, expression, and head pose and uses the FLAME head model Li et al. [2017] with predefined mesh topology and blendshapes learned from 3D scans. We use the face shape from the source image $\mathbf{x}_s$ as well as the facial expression and the camera pose from the target image $\mathbf{x}_t$ for further processing.

Head mesh reconstruction. The vertices of the DECA mesh (initial mesh) with personalized face region and generic non-facial parts are rendered into an xyz-coordinate texture using the predefined texture mapping. The xyz-texture and the neural texture $\mathbf{T}_s$ are concatenated and processed with the U-Net network Ronneberger et al. [2015] E_{geom} into a new texture map (*latent geometry* map) $\mathbf{Z}_t$. The 3D displacements for each mesh vertex are then decoded independently by the multi-layer perceptron G_{geom} that predicts a 3D offset $\Delta\hat{\mathbf{v}}$ for each vertex along the surface normal. This step reconstructs the personalized model for non-face parts of the head mesh. The reconstructions are compatible with the topology/connectivity of the FLAME mesh Li et al. [2017].

Deferred neural rendering. The personalized head mesh is rendered using the pose estimated by DECA for the target image and with the superimposed neural texture. The resulting render concatenated with rasterized surface normal and processed by the decoding (rendering) U-Net network E_{img} to obtain the predicted image $\hat{\mathbf{x}}_t$ and the segmentation mask $\hat{\mathbf{s}}_t$. During training, the reconstruction is compared to the true image/mask, and the losses are used to update the components of our system.

Below, we first provide details of training process and the training losses. We also discuss how a model with simplified inference and parameterization, which models head geometry using linear blendshape basis, can be trained.

7.3.2 Parametric face modeling

Our method uses a predefined head mesh with corresponding texture coordinates w. Also, our mesh reconstruction process does not change the head topology or texture coordinates of individual vertices. More specifically, we use the FLAME Li et al. [2017] head model that has N base vertices $\mathbf{v}_{\text{base}} \in \mathbb{R}^{3N}$, and two sets of K and L basis vectors that encode shape $\mathcal{B} \in \mathbb{R}^{3N\times K}$ and expression $\mathcal{D} \in \mathbb{R}^{3N\times L}$. The reconstruction is carried out in two stages: the basis vectors are first blended using the estimated vectors of linear coefficients ϕ and ψ, and then the linear blend skinning Li et al. [2017] function $\mathcal{W}$ is applied with estimated parameters θ, which rotates groups of vertices around linearly estimated joints. The final reconstruction in world coordinates can be expressed as follows:

$$\mathbf{v}(\phi,\psi,\theta) = \mathcal{W}\Big(\mathbf{v}_{\text{base}} + \mathcal{B}\phi + \mathcal{D}\psi,\ \theta\Big).$$

In previous works Thies et al. [2016a], these parameters were estimated via photometric optimization. More recently, learning-based methods Feng et al. [2020] capable of single-view reconstruction started to emerge. In our work, we use a pre-trained DECA system Feng et al. [2020] that provides an initial head reconstruction (in the form of FLAME parameters).

During training, we apply DECA to both source image $\mathbf{x}_s$ and the target image $\mathbf{x}_t$. The face shape parameters from the source image $\mathbf{x}_s$ alongside the expression, head pose and camera pose parameters from the target image $\mathbf{x}_t$ are then used to reconstruct the initial FLAME vertices $\mathbf{v}_t = \mathbf{v}(\phi_s, \psi_t, \theta_t)$, as well as estimate the camera matrix $\mathcal{P}_t$.

7.3.3 Head mesh reconstruction

The FLAME vertices $\mathbf{v}_t$ estimated by DECA provide good reconstructions for the face region but lack any person-specific details in the remaining parts of the head (hair, clothing). To alleviate that, we predict person-specific mesh offsets for non-facial regions while preserving the face shape predicted by DECA. We additionally exclude ear regions since their geometry in the initial mesh is too complex to be learned from in-the-wild video datasets.

These mesh offsets are estimated in two steps. First, we encode both vertex texture and the neural texture $\mathbf{T}_s$ into the latent geometry texture map $\mathbf{Z}_t$ via a UNet network E_{geom}. It allows the produced latent map to contain both positions of the initial vertices $\mathbf{v}_t$, and their semantics, provided by the neural texture.

From $\mathbf{Z}_t$ we obtain the vectors $\mathbf{Z}_t^w$ by bilinear interpolation at fixed texture coordinates w. The vectors $\mathbf{Z}_t^w$ and their coordinates w are then concatenated and passed through a multi-layer perceptron G_{geom} independently for each vertex in the mesh to predict the offsets $\Delta\hat{\mathbf{v}}_t$ and push them along normal $\vec{n}_t$. These displacements are then zeroed out for face and ear regions, and the final reconstruction in world coordinates is obtained as follows: $\hat{\mathbf{v}}_t = \mathbf{v}_t + \Delta\hat{\mathbf{v}}_t \cdot \vec{n}_t$.

7.3.4 Deferred neural rendering

We render the reconstructed head vertices $\hat{\mathbf{v}}_t$ using the topology and texture coordinates w from the FLAME model with the superimposed neural texture $\mathbf{T}_s$. For that, we use a differentiable mesh renderer $\mathcal{R}$ Ravi et al. [2020] with the camera matrix $\mathcal{P}_t$ estimated by DECA for the target image $\mathbf{x}_t$.

The resulting rasterization (neural texture and surface normal) is processed by the rendering (decoding) network E_{img} to obtain the predicted image $\hat{\mathbf{x}}_t$ and the segmentation mask $\hat{\mathbf{s}}_t$. E_{img} consists of two UNets that separately decode an image and a mask. The result of the deferred neural rendering is the reconstruction of the target image $\hat{\mathbf{x}}_t$ and its mask $\hat{\mathbf{s}}_t$, which is compared to the ground-truth image $\mathbf{x}_t$ and mask $\mathbf{s}_t$ via a photometric loss.

7.3.5 Training objectives

In our approach, we learn geometry without any ground-truth 3D supervision during training or pre-training (on top of the pretrained DECA estimator). For that we utilize two types of objectives: segmentation-based geometric losses $\mathcal{L}_{\text{geom}}$ and photometric losses $\mathcal{L}_{\text{photo}}$.

We found that explicitly assigning subsets of mesh vertices to the neck and the hair regions helps a lot with the quality of final deformations. It allows us to introduce a topological prior for the predicted offsets. In our predictions, hair has no holes and is topologically equivalent to a half-sphere (a disk), while neck vertices are equivalent to a cylinder.

To evaluate the geometric losses, we calculate two separate occupancy masks using a soft rasterization operation Liu et al. [2019]. First, $\hat{\mathbf{o}}_t^{\text{hair}}$ is calculated with detached neck vertices, so that the gradient flows through that mask only to the offsets corresponding to the hair vertices, and then $\hat{\mathbf{o}}_t^{\text{neck}}$ is calculated with detached hair vertices. We match the hair occupancy mask to the ground-truth mask $\mathbf{s}_t^{\text{hair}}$ (which covers the hair, face, and ears), and the neck occupancy mask to the whole segmentation mask $\mathbf{s}_t$: $\mathcal{L}_{\text{occ}} = \lambda_{\text{hair}} \left\| \hat{\mathbf{o}}_t^{\text{hair}} - \mathbf{s}_t^{\text{hair}} \right\|_2^2 + \lambda_{\text{neck}} \left\| \hat{\mathbf{o}}_t^{\text{neck}} - \mathbf{s}_t \right\|_2^2$.

We also use an auxiliary Chamfer loss to ensure that the predicted mesh vertices cover the head more uniformly. Specifically, we match the 2D coordinates of the mesh vertices, projected into the target image, to the head segmentation mask. We denote the subset of predicted mesh vertices, visible in the target image, as $\hat{\mathbf{p}}_t = \mathcal{P}'_t(\hat{\mathbf{v}}_t)$, and the number of these vertices as N_t, so that $\hat{\mathbf{p}}_t \in \mathbf{R}^{N_t \times 2}$. Notice that operator $\mathcal{P}'_t$ here not only does the camera transformation, but also discards the z coordinate of the projected mesh vertices. To compute the loss, we then sample N_t 2D points from the segmentation mask $\mathbf{s}_t$ and estimate the Chamfer distance between the sampled set of points $\mathbf{p}_t$ and the vertex projections:

$$\begin{aligned}\mathcal{L}_{\text{chm}} = &\frac{1}{2N_t} \sum_{\hat{p}_t \in \hat{\mathbf{p}}_t} \left\| \hat{p}_t - \arg\min_{p \in \mathbf{p}_t} \left\| p - \hat{p}_t \right\| \right\| + \\ &\frac{1}{2N_t} \sum_{p_t \in \mathbf{p}_t} \left\| p_t - \arg\min_{\hat{p} \in \hat{\mathbf{p}}_t} \left\| \hat{p} - p_t \right\| \right\|.\end{aligned}$$

Lastly, we regularize the learned geometry using the Laplacian penalty Sorkine-Hornung [2005]. We initially found that regularizing offsets $\Delta\hat{\mathbf{v}}$ worked better than regularizing full coordinates $\hat{\mathbf{v}}$ and stuck to that approach for all experiments. Our version of the Laplacian loss can be written as:

$$\mathcal{L}_{\text{lap}} = \frac{1}{V} \sum_{i=1}^{V} \left\| \Delta\hat{\mathbf{v}}_i - \frac{1}{\mathcal{N}(i)} \sum_{j \in \mathcal{N}(i)} \Delta\hat{\mathbf{v}}_j \right\|_1,$$

where $\mathcal{N}(i)$ denotes a set indices for vertices adjacent to the i-th vertex in the mesh.

The final geometric loss that we use to learn head mesh reconstruction is: $\mathcal{L}_{\text{geom}} = \mathcal{L}_{\text{occ}} + \lambda_{\text{chm}}\mathcal{L}_{\text{chm}} + \lambda_{\text{lap}}\mathcal{L}_{\text{lap}}$.

We also use photometric optimization that matches the obtained and the ground truth images. We have found that photometric loss terms not only allow to obtain photorealistic renders but also aid in learning proper geometric reconstructions. The photometric terms include the perceptual loss Johnson et al. [2016b], the face recognition loss Cao et al. [2018b] and the multi-resolution adversarial loss Goodfellow et al. [2014b], Wang et al. [2018f]. We also use the Dice loss Milletari et al. [2016] to match segmentation masks. Therefore, we use the following combination of losses: $\mathcal{L}_{\text{photo}} = \lambda_{\text{per}}\mathcal{L}_{\text{per}} + \lambda_{\text{idt}}\mathcal{L}_{\text{idt}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{seg}}\mathcal{L}_{\text{seg}}$.

The final objective is a sum of the geometric and the photometric losses: $\mathcal{L} = \mathcal{L}_{\text{geom}} + \mathcal{L}_{\text{photo}}$.

7.3.6 Linear deformation model

In addition to our full non-linear model introduced above, we also consider a simplified parametric model with a linear basis of offsets $\Delta\hat{\mathbf{v}}$. While this model is similar to parametric models Li et al. [2017], Zuffi et al. [2019], we still do not use 3D scans for training and rather obtain our linear model by "distilling" our non-linear model. We also train a feedforward regressor that predicts the linear coefficients from an input image.

The motivation for training this additional model is to show that the deformations learned by our method can be approximated using a system with a significantly lower capacity. Such a simple regression model can be easier to apply for inference on low-performance devices.

To train the linear model, we first obtain the basis of offsets $\mathcal{F} \in \mathbb{R}^{3N\times K}$, which is similar to the ones used in the FLAME parametric model. This basis is obtained by applying a low-rank PCA Halko et al. [2011] to the matrix of offsets $\Delta\hat{\mathbf{V}} \in \mathbb{R}^{3N\times M}$, calculated using M images from the dataset. Following Li et al. [2017], we discard most of the basis vectors and only keep K components corresponding to maximal singular values. The approximated vertex offsets $\tilde{\mathbf{v}}$ for each image can then be estimated as following $\tilde{\mathbf{v}} = \mathcal{F}\eta$, where η can be obtained by applying the pseudo-inverse of a basis matrix $\mathcal{F}$ to the corresponding offsets $\Delta\hat{\mathbf{v}}$: $\eta = \left(\mathcal{F}^T\mathcal{F}\right)^{-1}\mathcal{F}^T\Delta\hat{\mathbf{v}}$

We then train the regression network by estimating a vector of basis coefficients η_t, given an image $\mathbf{x}_t$. For that, we minimize the mean square error (MSE) loss $\|\hat{\eta}_t - \eta_t\|_2^2$ between the estimated coefficients and the ground truth.

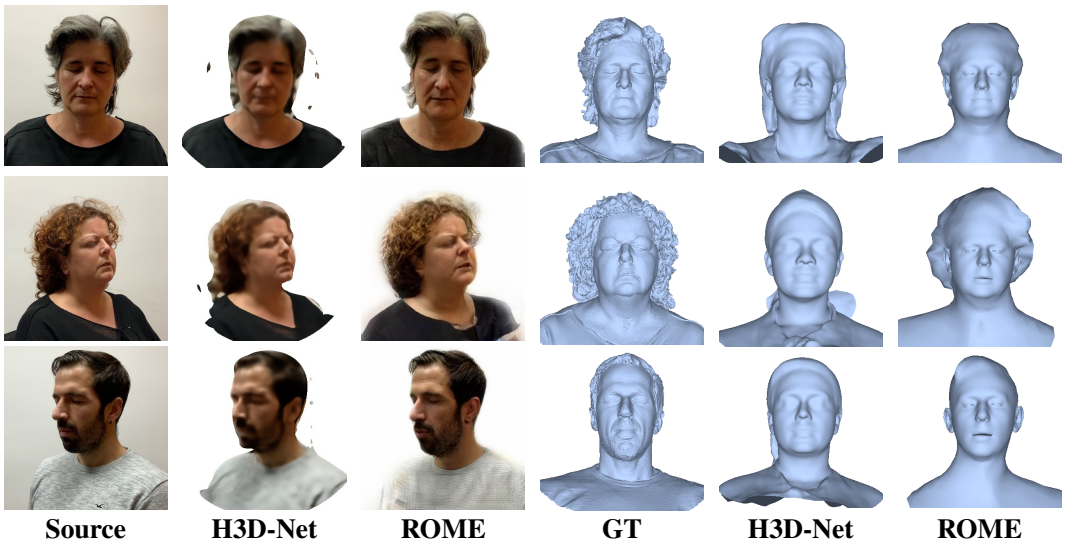


FIGURE 7.3: Qualitative comparison on representative cases for the H3DS dataset. While neither of the two methods achieve perfect results, arguably, ROME achieves more realistic renders, as well as better matches the head geometry than H3D-Net in the single-shot mode. An important advantage of ROME is that the resulting avatars are ready for animation.

7.4 Experiments

We train our models on the VoxCeleb2 Chung et al. [2018b] dataset of videos. This large-scale dataset contains an order of 10^5 videos of 10^3 different speakers. It is widely used Doukas et al. [2021a], Wang et al. [2021], Zakharov et al. [2020] to train talking head models. However, the main drawback of this dataset is the mixed quality of videos and the heavy bias towards frontal poses.

To address these well-known limitations, we process this dataset using an off-the-shelf image quality analysis model Su et al. [2020a] and a 3D face-alignment network Bulat and Tzimiropoulos [2017b]. We then filter out the data which has poor quality and non-diverse head rotations. Our final training dataset has ≈ 15000 sequences. We note that filtering/pruning does not fully solve the problem of head rotation bias, and our method still works best in frontal views. For more details, please refer to the supplementary materials.

We also use the H3DS Ramon et al. [2021] dataset of photos with associated 3D scans to evaluate the quality of head reconstructions.

7.4.1 Implementation details

In the experiments, unless noted otherwise, we train all architectures jointly and end-to-end. We use the following weights: $\lambda_{\text{hair}} = 10$, $\lambda_{\text{per}} = 1$, $\lambda_{\text{idt}} = 0.1$, $\lambda_{\text{adv}} = 0.1$, $\lambda_{\text{seg}} = 10$, ($\lambda_{\text{hair}} = 10$,

TABLE 7.1: Evaluation results on the H3DS dataset in one-shot scenario for our models, H3D-Net, and DECA. We compute Chamfer distance (lower is better) across all available scans, reconstructed from three different viewpoints. ROME variants significantly exceeds H3D-Net in the one-shot reconstruction quality.

Method	DECA	H3D-Net	ROME	LinearROME
Chamfer Distance	14.98	15.1	12.6	12.45

$\lambda_{\text{neck}} = 1$) and enable the neck and the 2D Chamfer loss $\lambda_{\text{chm}} = 0.01$) and $\lambda_{\text{lap}} = 10$. We ablate all geometry losses and method parts below.

We demonstrate results of ablation study at Figure 7.6. As expected, predicting more accurate geometry affect the renders (first row). Also, we verify the necessity of all terms of geometry loss.

We train our models at 256×256 resolution using ADAM Kingma and Ba [2015] with the fixed learning rate of 10^{-4}, $\beta_1 = 0$, $\beta_2 = 0.999$, and a batch size of 32. For more details, please refer to the supplementary materials.

7.4.2 Evaluation

3D reconstruction. We evaluate our head reconstruction quality using a novel H3DS dataset Ramon et al. [2021]. We compare against the state-of-the-art head reconstruction method H3D-Net Ramon et al. [2021], which uses signed distance functions to represent the geometry. While providing great reconstruction quality in the sparse-view scenario, their approach has several limitations. For example, H3D-Net requires a dataset of full head scans to learn the prior on head shapes. Additionally, its results do not have fixed topology or rigging and their method requires fine-tuning per scene, while our method works in a feed-forward way.

We carry out the comparison with H3D-Net in a single-view scenario, which is native for our method but is beyond the capabilities stated by the authors in the original publication Ramon et al. [2021]. However, to the best of our knowledge, H3D-Net is the closest system to ours in single-view reconstruction capabilities (out of all systems with either their code or results available). Additionaly, we tried to compare our system with PIFuHD Saito et al. [2020a], which unfortunately failed to work with heads images without body (see supplementary).

We show qualitative comparison in Figure 7.3. We evaluate our method and H3D-Net both for frontal- and side-view reconstruction. We note the significant overfitting of H3D-Net to the visible hair geometry, while our model provides reconstructions more robust to the change of viewpoint.

FIGURE 7.4: Comparison of renders on a VoxCeleb2 dataset. The task is to reenact the **source** image with the expression and pose of the **driver** image. Here, we picked diverse examples in terms of pose variation to highlight the differences in performance of compared methods. We observe that for the large head pose rotations, purely neural-based methods (**FOMM**, **Bi-Layer**) struggle to maintain consistent quality. In comparison, our rendering method produces images that are more robust to pose changes. Admittedly, for small pose changes, neural-based methods exhibit a smaller identity gap than ROME (bottom row) and overall outperform our method in terms of rendering quality. Additionally, we include a **FLAMETex** method, which is employed in state-of-the-art one-shot face reconstruction systems Feng et al. [2020] but is not able to personalize the avatar at the head level.

In total, we compared our models on all scans available in the test set of the H3DS dataset, and each scan was reconstructed from three different viewpoints. We provide the measured mean Chamfer distance both for our method and baselines across all scans in Tab. 7.1.

Rendering. We evaluate the quality of our renders on the hold-out subset VoxCeleb2 dataset. We use a cross-driving comparison scenario for qualitative comparison to highlight the animation capabilities of our method, and self-driving scenario for quantitative comparison.

First, we compare with a FLAMETex Li et al. [2017] rendering system, which works explicitly with mesh rendering. From the source image, FLAMETex estimates the albedo via a basis of RGB textures, and then combines it with predicted scene-specific shading. In contrast, our method predicts a rendered image directly and avoids the complexity of explicit albedo-shading decomposition.

We then compare with publicly available geometry-free rendering methods, which were trained on the same dataset. For that, we use the First-Order Motion Model (FOMM) Siarohin et al. [2019b] and the Bi-Layer Avatar Model Zakharov et al. [2020]. Both these systems bypass explicit 3D geometry estimation and rely only on learning the scene structure via the parameters of generative ConvNets. Other methods Doukas et al. [2021a], Wang et al. [2021], which internally utilize

TABLE 7.2: Here we present the quantitative results on the VoxCeleb2 dataset in the self-reenactment and cross-reenactment modes. Our ROME system performs on par with FOMM in self-reenactment, but outperforms (in the most perceptually-plausible LPIPS metrics) On the contrary, in cross-driving, when the task is complex for pure neural based, our method obtains better results.

	self-reenactment			cross-reenactment		
Method	LPIPS↓	SSIM↑	PSNR↑	FID↓	CSIM↑	IQA↑
FOMM	0.09	0.87	25.8	52.95	0.53	55.9
Bi-Layer	0.08	0.83	23.68	51.35	0.56	50.48
TPSMM	0.09	0.85	26.1	49.27	0.57	59.5
ROME	0.08	0.86	26.2	45.32	0.62	66.3

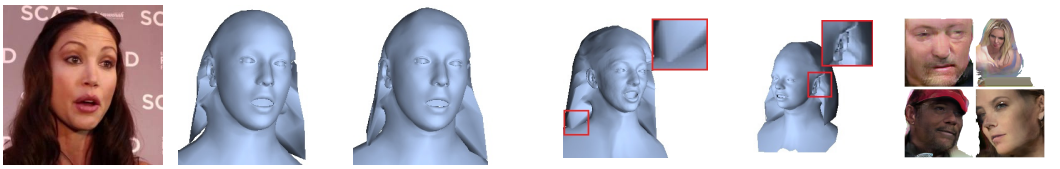
some 3D structures, like camera rotations, were out of the scope of our comparison due to the unavailability of pre-trained models.

We present the qualitative comparison in Figure 7.4, and a quantitative comparison across a randomly sampled hold-out VoxCeleb2 subset in Table 7.2. We restrict the comparison to the face and hair region as the shoulder pose is not controlled by our method (driven by DECA parameters), which is admittedly a limitation of our system. We thus mask the results according to the face and hair mask estimated from the ground truth image.

Generally, we observe that over the entire test set, the quality of ROME avatars in the self-reenactment mode is similar to FOMM and better than the Bi-layer model. For the cross-reenactment scenario, our model is clearly better than both alternatives according to three metrics, that help to asses unsupervised quality of the images in three aspects: realism, identity preservation and blind quality of the image. The huge gap for IQA Su et al. [2020b] and FID Heusel et al. [2017b] is also noticeable in the qualitative comparison, especially for strong pose change (see CSIM Zakharov et al. [2019] column in Tab. 7.2). The PSNR and SSIM metrics penalize slight misalignments between the sharp ground truth and our renderings much stronger than the blurriness in FOMM reconstructions. The advatage of ROME avatar is noticable even for self-driving case according to LPIPS. We provide a more extensive qualitative evaluation in the supplementary materials.

7.4.3 Linear basis experiments

As discussed above, we distill our ROME head reconstruction model into a linear parametric model. To do that, we set the number of basis vectors to 50 for the hair and 10 for the neck offsets and run low-rank Principle Component Analysis (PCA) to estimate them. The number of components is chosen to obtain a low enough approximation error. Interestingly, the offsets learned by our model can be compressed by almost two orders of magnitude in terms of degrees of freedom without any practical loss in quality (Figure 7.5a), which suggests that the capacity



(A) Linear model

(B) Limitations

FIGURE 7.5: Linear model results and the examples of limitations. Offsets learned by our method (**ROME**) could be distilled using a linear parametric model. We are able to compress the offsets learned by our method into a small basis (**PCA Rec.**), reducing the degrees of freedom by two orders of magnitude. We can then **distill** these offsets using a much faster regression network with a small gap in terms of the reconstruction quality. The right image highlights the main limitations of our method, which include the failure to model long hair due to incorrect prior with small details (left), no coverage of ears (middle) and unrealistic renders under strong scale change (right).

of the offset generator is underused in our model. We combine estimated basis vectors with the original basis of the FLAME.

After that, we train feed-forward encoders that directly predict the coefficients of the two basis from the source image. The prediction is performed in two stages. First, face expression, pose and camera parameters are predicted with a MobileNetV2 Sandler et al. [2018] encoder. Then a slower ResNet-50 encoder He et al. [2016a] is used to predict hair, neck and shape coefficients. The choice of architectures are motivated by the fact that in many practical scenarios only the first encoder needs to be invoked frequently (per-frame), while the second can run at much lower rate or even only at the model creation time.

7.4.4 Limitations

In some cases, the proposed system often produces somewhat oversmoothed geometry without person-specific attributes. Most of those attributes simply cannot be presented with the limited set of vertices. Our preliminary experiments with subdivision during training did not help to alleviate this problem. It may be interesting to study this problem in a few-shot scenario (given different views) or to use images and proxy geometry in a higher resolution. Another frequent geometry artifact is the roughness of the clothing approximation. As we know, modeling any clothes without 3D data is an extremely difficult task and the fact that our models is not able to do so from videos is not surprising. A principled solution to this problem will likely require learning from datasets that cover these areas (e.g., circular 3D scans of human heads) Alldieck et al. [2019].

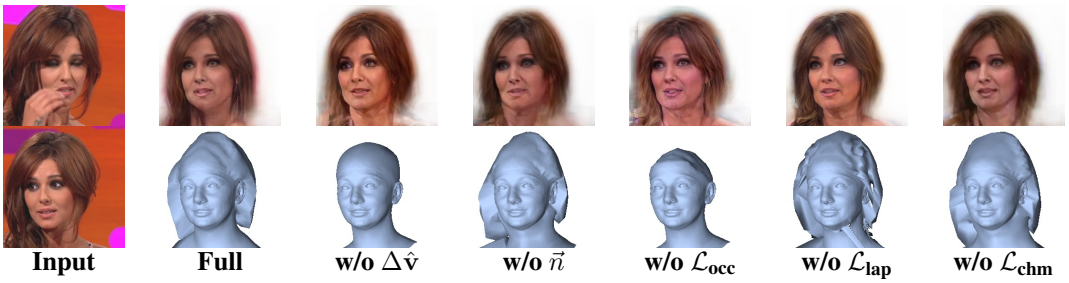


FIGURE 7.6: Ablation study. We qualitatively evaluate the individual components of our *full* model. w/o $\Delta\hat{\mathbf{v}}$: without the per-vertex displacements, we obtain a significantly worse render quality. w/o $\vec{n}$: when we apply per-vertex deformations instead of per-vertex displacements (i.e., deformations alongside the normals), we obtain noisy reconstructions (especially in the area of the neck) and worse renders. w/o $\mathcal{L}_{\text{occ}}$: without silhouette-based losses, our model fails to learn proper reconstructions. w/o $\mathcal{L}_{\text{lap}}$: Laplacian regularization smooths the reconstructions and reduces noise while not affecting the quality of renders. w/o $\mathcal{L}_{\text{chm}}$: chamfer loss allows us to constrain the displaced vertices to lie inside the scene boundaries, which positively affects the smoothness of the visible part of the reconstruction.

Our current model is trained at roughly fixed scale, though explicit geometry modeling allows it to generalize to adjacent scale reasonably well. Still, strong changes of scale lead to poor performance (Figure 7.5b). More examples are provided in the supplementary materials. Addressing this issue via mip-mapping and multi-scale GAN training techniques remains future work.

Lastly, our model can have artifacts with long hair (Figure 7.5b, left) or ears (Figure 7.5b, middle). Handling such cases gracefully are likely to require a departure from the predefined FLAME mesh connectivity to new person-specific mesh topology. Handling such issues using a limited set of pre-designed hair meshes is an interesting direction for future research.

7.5 Summary

We have presented ROME avatars: a system for creating realistic one-shot mesh-based human head models that can be animated and are compatible with FLAME head models. We compare our model with representative state-of-the-art models from different classes, and show that it is highly competitive both in terms of geometry estimation and the quality of rendering.

Crucially, our system can learn to model head geometry without direct supervision in the form of 3D scans. Despite that, we have observed it to achieve state-of-the-art results in head geometry recovery from a single photograph. At the same time, it also performs better than previous one-shot neural rendering approaches in the cross- and self-driving scenario. We have thus verified that the resulting geometry could be used to improve the rendering quality.

As neural rendering becomes more widespread within graphics systems, ROME avatars and similar systems can become directly applicable, while their one-shot capability and the simplicity

of rigging derived from DECA and FLAME could become especially important in practical applications.

Chapter 8

Conclusion

In this thesis, the problem of synthesis of the human face and body images has been extensively explored from computer vision and computer graphics perspectives. The works which constitute this thesis advanced the human image synthesis systems in the following ways: proposed one of the first methods of realistic synthesis of human heads by leveraging large-scale training techniques; reduced the computational time required for initialization and inference of such systems and improved their quality to achieve high realism; proposed an efficient way of disentangling motion and appearance signals; and finally contributed to the task of one-shot full 3D rigged human head reconstruction. Crucially, in all the proposed systems generative modeling approach remains the key to achieving highly realistic synthesis.

We presented a new approach to GAN-based training for semantic image editing, one of the fundamental problems in human image synthesis. Then, we have explored geometry-free approaches for human image synthesis based on the either direct prediction of the outputs using convolutional neural networks, warping-based approaches that utilize an RGB texture, and hybrid approaches that combine the two to achieve significant rendering speed-ups. Lastly, we have proposed two approaches for incorporating 3D constraints into the geometry-free models. One system utilizes a mesh representation and is capable of one-shot coarse reconstruction and realistic rendering. Another model utilizes volumetric modeling to improve the performance of one-shot rendering systems in the challenging cross-reenactment scenario.

The works on which this thesis is based have contributed not only to the scientific advancements in the area of neural avatars but also to the public domain. Specifically, the work of Chapter 4, sparked a discussion in both the mainstream and social media regarding the potential misuse of the so-called “deep fake” systems, that allow the creation of fake media with human subjects. It has also been discussed in publications by BBC, CNN, The Telegraph, and other news outlets. This level of public attention has resulted in the publication being ranked by Altmetric rating as the most discussed scientific work of 2019 Alt.

While the potential for misuse of these early systems at a time was quite low due to the low quality of the results, modern systems such as the ones described in Chapter 6 in conjunction with large-scale text and image generative models such as GPT-3 Brown et al. [2020] or Latent Diffusion Rombach et al. [2021] can produce increasingly sophisticated and believable synthetic media. While the technology of human avatars is certainly disruptive and has numerous beneficial applications in the areas of special effects, image generation, and editing, the potential for misuse of these technologies is quite high and remains the focus of research Dolhansky et al. [2020], Rössler et al. [2019].

Further directions on human avatar synthesis may revolve around the following problems. While neural rendering has achieved exceptionally high degrees of realism, it cannot benefit from the efficient classical rendering techniques developed in computer graphics. Bridging these two areas together by making the realistic human rendering systems rely heavily on classical rendering and physics modeling is an exciting new direction for research.

Another direction could be formulated as further incorporating the geometric constraints into the geometry-free systems. Such constraints help improve the robustness of such systems to the simple geometric transformations, such as camera rotation and translation, which are fundamental to such applications as telepresence.

Bibliography

Altmetric top 100. https://www.altmetric.com/top100/2019/.

Avatar SDK homepage. https://avatarsdk.com.

Pytorch. http://pytorch.org/.

SNPE homepage. https://developer.qualcomm.com/sites/default/files/docs/snpe.

TensorFlow Lite homepage. https://www.tensorflow.org/lite.

Kfir Aberman, Mingyi Shi, Jing Liao, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Deep video-based performance cloning. *arXiv preprint arXiv:1808.06847*, 2018.

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, 2010a.

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, 2010b.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018a.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *Proc. CVPR*, June 2018b.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *2018 International Conference on 3D Vision (3DV)*, pages 98–109. IEEE, 2018c.

Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, 2016.

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.

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.

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proc. ICML*, pages 214–223, 2017.

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.

Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. Neural codes for image retrieval. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 584–599, 2014.

Guha Balakrishnan, Amy Zhao, Adrian V. Dalca, Frédo Durand, and John V. Guttag. Synthesizing images of humans in unseen poses. In *Proc. CVPR*, pages 8340–8348, 2018.

Alexandru O Bălan and Michael J Black. The naked truth: Estimating body shape under clothing. In *Proc. ECCV*, pages 15–29. Springer, 2008.

Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. In *Proc. NIPS*, pages 752–762, 2017.

Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99*, 1999.

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

Volker Blanz, Sami Romdhani, and Thomas Vetter. Face identification across different poses and illuminations with a 3d morphable model. *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 202–207, 2002.

Federica Bogo, Michael J Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *Proc. ICCV*, pages 2300–2308, 2015.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Proc. ECCV*, pages 561–578. Springer, 2016.

Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *CoRR*, abs/1609.07093, 2016.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019a.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019*, 2019b.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

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, 2017a.

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 *International Conference on Computer Vision*, 2017b.

Egor Burkov, I. Pasechnik, Artur Grigorev, and Victor S. Lempitsky. Neural head reenactment with latent pose descriptors. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13783–13792, 2020.

Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. *arXiv preprint arXiv:1806.08472*, 2018a.

Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74, 2018b.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. CVPR*, 2017.

Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4d video textures for interactive character appearance. In *Computer Graphics Forum*, volume 33, pages 371–380. Wiley Online Library, 2014.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. *arXiv preprint arXiv:1808.07371*, 2018.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. ECCV*, 2018.

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proc. ICCV*, pages 1520–1529, 2017.

Soumith Chintala, Emily Denton, Martin Arjovsky, and Michael Mathieu. How to train a GAN? Tips and tricks to make GANs work. https://github.com/soumith/ganhacks, 2017.

Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. CVPR*, 2018.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018a.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018b.

Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4):69, 2015.

Paul E. Debevec, Yizhou Yu, and George Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Rendering Techniques '98, Proceedings of the Eurographics Workshop in Vienna, Austria, June 29 - July 1, 1998*, pages 105–116, 1998.

Paul E. Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.

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

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

Jiankang Deng, J. Guo, Evangelos Ververas, Irene Kotsia, Stefanos Zafeiriou, and InsightFace FaceSoft. Retinaface: Single-shot multi-level face localisation in the wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5202–5211, 2020.

Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv: Computer Vision and Pattern Recognition*, 2020.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Proc. ECCV*, pages 184–199, 2014.

Craig Donner, Tim Weyrich, Eugene d'Eon, Ravi Ramamoorthi, and Szymon Rusinkiewicz. A layered, heterogeneous reflectance model for acquiring and rendering human skin. In *ACM Transactions on Graphics (TOG)*, volume 27, page 140. ACM, 2008.

Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. NIPS*, pages 658–666, 2016.

Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proc. CVPR*, pages 1538–1546, 2015.

Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: real-time volumetric performance capture. *ACM Transactions on Graphics (TOG)*, 36(6):246, 2017.

Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: Video-and-audio-driven talking head synthesis. 2021a.

Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. 2021b.

Bernhard Egger, W. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhöfer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39:1 – 38, 2020.

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2009.

FaceApp. Faceapp. https://www.faceapp.com/, 2018.

Andrew Feng, Dan Casas, and Ari Shapiro. Avatar reshaping and automatic rigging using a deformable model. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 57–64. ACM, 2015.

Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018.

Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (TOG)*, 40:1 – 13, 2020.

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

Tobias Fischer, Hyung Jin Chang, and Y. Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *ECCV*, 2018.

Chaoyou Fu, Yibo Hu, Xiang Wu, Guoli Wang, Qian Zhang, and Ran He. High fidelity face manipulation with extreme pose and expression. *arXiv preprint arXiv:1903.12003*, 2019.

Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. pages 8645–8654, 2021.

Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *Proc. ECCV*, pages 311–326. Springer, 2016.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, 2016.

Dariu M. Gavrila and Larry S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

Bastian Goldlücke and Daniel Cremers. Superresolution texture maps for multiview reconstruction. In *Proc. ICCV*, pages 1677–1684, 2009.

Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, M. Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7442–7451, 2019a.

Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019b.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014a.

Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014b.

Riza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. DenseReg: Fully convolutional dense shape regression in-the-wild. In *Proc. CVPR*, volume 2, page 5, 2017.

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense human pose estimation in the wild. In *Proc. CVPR*, June 2018.

Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. *CoRR*, abs/1911.08139, 2019.

Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *AAAI*, 2020.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53: 217–288, 2011.

Nils Hasler, Hanno Ackermann, Bodo Rosenhahn, Thorsten Thormählen, and Hans-Peter Seidel. Multilinear pose and body shape estimation of dressed subjects from image sets. In *Proc. CVPR*, pages 1823–1830. IEEE, 2010.

Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4295–4304, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015a. URL http://arxiv.org/abs/1512.03385.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International*

Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, pages 1026–1034, 2015b.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference*, 2016b.

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., 2017a.

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 *Advances in Neural Information Processing Systems*, 2017b.

Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.*, 36(6):195:1–195:14, 2017.

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

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

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4):107:1–107:14, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

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.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proc. NIPS*, pages 2017–2025, 2015.

Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *Proc. NIPS*, pages 769–776, 2009.

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.

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.

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

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

Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *7th International Conference on Learning Representations, ICLR*, 2019.

Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Takeo Kanade, Peter Rander, and P. J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multim.*, 4:34–47, 1997.

Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proc. CVPR*, 2018.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019.

Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson W.H. Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *AAAI*, 2022.

Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan P. Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4285–4295, 2021.

Dong-Wook Kim, Jae Ryun Chung, and Seung-Won Jung. GRDN: grouped residual dense network for real image denoising and gan-based real-world noise modeling. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019*, 2019.

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*, 2018a.

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

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

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proc. CVPR*, pages 1646–1654, 2016.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

Oliver Klehm, Fabrice Rousselle, Marios Papas, Derek Bradley, Christophe Hery, Bernd Bickel, Wojciech Jarosz, and Thabo Beeler. Recent advances in facial appearance capture. In *Computer Graphics Forum*, volume 34, pages 709–733. Wiley Online Library, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324, 1998.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017a.

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. CVPR*, 2017b.

Victor S. Lempitsky and Denis V. Ivanov. Seamless mosaicing of image-based texture maps. In *Proc. CVPR*, 2007.

Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (TOG)*, 36:1 – 17, 2017.

Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. Deep human parsing with active template regression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(12):2402–2414, Dec 2015a. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2408360.

Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Si Liu, Jinhui Tang, Liang Lin, and Shuicheng Yan. Human parsing with contextualized convolutional neural network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:115–127, 2015b.

Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural animation and reenactment of human actor videos. *arXiv preprint arXiv:1809.03658*, 2018.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7707–7716, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015.

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

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

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

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason M. Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40:1 – 13, 2021.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6): 248, 2015.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? A large-scale study. *CoRR*, abs/1711.10337, 2017.

Qianli Ma, Shunsuke Saito, Jinlong Yang, Siyu Tang, and Michael J. Black. Scale: Modeling clothed humans with a surface codec of articulated local elements. In *CVPR*, 2021.

Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proc. CVPR*, 2015.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016.

Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien P. C. Valentin, Sameh Khamis, Philip L. Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B. Goldman, Cem Keskin, Steven M. Seitz, Shahram Izadi, and Sean Ryan Fanello. *LookinGood*: enhancing performance capture with real-time neural re-rendering. *ACM Trans. Graph.*, 37(6):255:1–255:14, 2018.

Simon Osindero Mehdi Mirza. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016.

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

Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. GANerated hands for real-time 3d hand tracking from monocular RGB. In *Proc. CVPR*, June 2018.

Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. pagan: real-time avatars using dynamic textures. *ACM Trans. Graph.*, 37:258, 2019.

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

Natalia Neverova, Riza Alp Güler, and Iasonas Kokkinos. Dense pose transfer. In *Proc. ECCV*, September 2018.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *ArXiv*, abs/2104.10078, 2021.

Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. *2018 International Conference on 3D Vision (3DV)*, pages 484–494, 2018.

Jeong Joon Park, Peter R. Florence, Julian Straub, Richard A. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019.

Keunhong Park, U. Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. 2021a.

Keunhong Park, U. Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. 2021b.

O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. BMVC*, 2015a.

Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015b.

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proc. CVPR*, June 2018.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. 2009.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 1*, 2:559–572, 1901.

Pinscreen. `https://www.pinscreen.com/`.

Stylianos Ploumpis, Evangelos Ververas, Eimear O' Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick E. Pears, W. Smith, Baris Gecer, and Stefanos Zafeiriou. Towards a complete 3d morphable model of the human head. 2021.

Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)*, 34(4):120, 2015.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia Giraldez, Xavier Giró i Nieto, and Francesc Moreno-Noguer. H3d-net: Few-shot high-fidelity 3d head reconstruction. *ArXiv*, abs/2107.12512, 2021.

Alex Rav-Acha, Pushmeet Kohli, Carsten Rother, and Andrew W. Fitzgibbon. Unwrap mosaics: a new representation for video editing. *ACM Trans. Graph.*, 27(3):17:1–17:11, 2008.

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014*, pages 512–519, 2014.

Nadia Robertini, Dan Casas, Edilson De Aguiar, and Christian Theobalt. Multi-view performance capture of surface details. *International Journal of Computer Vision*, 124(1):96–113, 2017.

Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–11, 2019.

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

David E. Rumelhart, Geoffrey E. Hinton, and James L. McClelland. A general framework for parallel distributed processing. 1986a.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986b.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. URL http://arxiv.org/abs/1409.0575.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020a.

Shunsuke Saito, Tomas Simon, Jason M. Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 81–90, 2020b.

Mehdi S. M. Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proc. ICCV*, 2017.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proc. NIPS*, pages 2226–2234, 2016.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

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.

Terrence J. Sejnowski. The deep learning revolution. 2018.

Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proc. ECCV*, September 2018.

Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and Victor S. Lempitsky. Textured neural avatars. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proc. CVPR*, June 2018.

Aliaksandr Siarohin, Stéphane Lathuilière, S. Tulyakov, Elisa Ricci, and N. Sebe. Animating arbitrary objects via deep motion transfer. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2372–2381, 2019a.

Aliaksandr Siarohin, Stéphane Lathuilière, S. Tulyakov, Elisa Ricci, and N. Sebe. First order motion model for image animation. *ArXiv*, abs/2003.00196, 2019b.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, 2019c.

Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

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

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015b.

Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America. A, Optics and image science*, 4 3: 519–24, 1987.

Olga Sorkine-Hornung. Laplacian mesh processing. In *Eurographics*, 2005.

J Starck and A Hilton. Model-based multiple view reconstruction of people. In *Proc. ICCV*, pages 915–922, 2003.

Ian Stavness, C Antonio Sánchez, John Lloyd, Andrew Ho, Johnty Wang, Sidney Fels, and Danny Huang. Unified skinning of rigid and deformable models for anatomical simulations. In *SIGGRAPH Asia 2014 Technical Briefs*, page 9. ACM, 2014.

Shaolin Su, Qingsen Yan, Yu Zhu, Cui cui Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. Blindly assess image quality in the wild guided by a self-adaptive hyper network. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3664–3673, 2020a.

Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. 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)*, 2020b.

Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. 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)*, 2020c.

Diana Sungatullina, Egor Zakharov, Dmitry Ulyanov, and Victor Lempitsky. Image manipulation with perceptual discriminators. In *Proc. ECCV*, September 2018a.

Diana Sungatullina, Egor Zakharov, Dmitry Ulyanov, and Victor S. Lempitsky. Project webpage. http://egorzakharov.github.io/perceptual_gan, 2018b.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021.

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.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *CoRR*, abs/1611.02200, 2016. URL http://arxiv.org/abs/1611.02200.

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

Masanori Koyama Takeru Miyato. cgans with projection discriminator. *arXiv:1802.05637*, 2018.

Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proc. CVPR*, pages 103–110. IEEE, 2012.

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

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

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, 2016b.

Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *arXiv: Computer Vision and Pattern Recognition*, 2019a.

Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: real-time face capture and reenactment of rgb videos. *ArXiv*, abs/2007.14808, 2019b.

A. Tran, Tal Hassner, Iacopo Masi, and Gérard G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1493–1502, 2017.

Soumya Tripathy, Juho Kannala, and Esa Rahtu. Icface: Interpretable and controllable face reenactment using gans. *CoRR*, abs/1904.01909, 2019. URL http://arxiv.org/abs/1904.01909.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proc. CVPR*, June 2018.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proc. ICML*, pages 1349–1357, 2016.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. In *Proc. CVPR*, 2018.

Paul Upchurch, Jacob R. Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Q. Weinberger. Deep feature interpolation for image content changes. In *Proc. CVPR*, pages 6090–6099, 2017.

Marco Volino, Dan Casas, John P Collomosse, and Adrian Hilton. Optimal representation of multi-view video. In *Proc. BMVC*, 2014.

H. Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jin Zhou, and Wenyu Liu. Cosface: Large margin cosine loss for deep face recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018a.

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. *arXiv preprint arXiv:1711.11585*, 2017.

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*, 2018b.

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*, 2018c.

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

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*, 2018e.

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. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018f.

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*, 2018g.

Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, 2019.

Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10034–10044, 2021.

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, April 2004a. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861. URL http://dx.doi.org/10.1109/TIP.2003.819861.

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

Lingyu Wei, Liwen Hu, Vladimir Kim, Ersin Yumer, and Hao Li. Real-time hair rendering using sequential adversarial networks. In *Proc. ECCV*, September 2018.

Alexander Weiss, David Hirshberg, and Michael J Black. Home 3d body scans from noisy image and range data. In *Proc. ICCV*, pages 1951–1958. IEEE, 2011.

Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, et al. Analysis of human faces using a measurement-based skin reflectance model. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1013–1024. ACM, 2006.

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

Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proc. ICCV*, pages 3756–3764, 2015.

Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Alexander Hypes, Taylor Koska, Steven Krenn, Stephen Lombardi, Xiaomin Luo, Kevyn McPhail, Laura Millerschoen, Michal Perdoch, Mark Pitts, Alexander Richard, Jason Saragih, Junko Saragih, Takaaki Shiratori, Tomas Simon, Matt Stewart, Autumn Trimble, Xinshuo Weng, David Whitewolf, Chenglei Wu, Shoou-I Yu, and Yaser Sheikh. Multiface: A dataset for neural face rendering. In *arXiv*, 2022. doi: 10.48550/ARXIV.2207.11243. URL https://arxiv.org/abs/2207.11243.

Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. Video-based characters: creating new human performances from a multi-view video database. *ACM Transactions on Graphics (TOG)*, 30 (4):32, 2011.

Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. 2021.

Lingbo Yang, C. Liu, P. Wang, Shanshe Wang, P. Ren, Siwei Ma, and W. Gao. Hifacegan: Face renovation via collaborative suppression and replenishment. *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed A. Elgharib, Daniel Cremers, and Christian Theobalt. i3dmm: Deep implicit 3d morphable model of human heads. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12798–12808, 2021.

Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. Adversarial meta-learning. *CoRR*, abs/1806.03316, 2018. URL http://arxiv.org/abs/1806.03316.

Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proc. CVPR*, pages 7287–7296. IEEE Computer Society, 2018.

Ilya Zakharkin, Kirill Mazur, Artur Grigoriev, and Victor S. Lempitsky. Point-based modeling of human clothing. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor S. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. 2019.

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

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1612.03242, 2016a.

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.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, 2016b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018a.

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

Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference*, 2016.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV*, pages 2242–2251, 2017a.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017b.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017c.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Proc. NIPS*, pages 465–476, 2017d.

Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J. Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild", 2019.