Imperial College London
Department of Computing

Photo-realistic Face Synthesis and Reenactment with Deep Generative Models

Michail Christos Doukas

Submitted in part fulfilment of the requirements for the degree
of Doctor of Philosophy in Computing and the Diploma
of Imperial College London, September 2022

To my beloved parents and sister

I hereby declare that this thesis is entirely my own work, and except where otherwise indicated, describes my own research.

Michail Christos Doukas
September 2022

©

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

The advent of Deep Learning has led to numerous breakthroughs in the field of Computer Vision. Over the last decade, a significant amount of research has been undertaken towards designing neural networks for visual data analysis. At the same time, rapid advancements have been made towards the direction of deep generative modeling, especially after the introduction of Generative Adversarial Networks (GANs), which have shown particularly promising results when it comes to synthesising visual data. Since then, considerable attention has been devoted to the problem of photo-realistic human face animation due to its wide range of applications, including image and video editing, virtual assistance, social media, teleconferencing, and augmented reality.

The objective of this thesis is to make progress towards generating photo-realistic videos of human faces. To that end, we propose novel generative algorithms that provide explicit control over the facial expression and head pose of synthesised subjects. Despite the major advances in face reenactment and motion transfer, current methods struggle to generate video portraits that are indistinguishable from real data. In this work, we aim to overcome the limitations of existing approaches, by combining concepts from deep generative networks and video-to-video translation with 3D face modelling, and more specifically by capitalising on prior knowledge of faces that is enclosed within statistical models such as 3D Morphable Models (3DMMs). In the first part of this thesis, we introduce a person-specific system that performs full head reenactment using ideas from video-to-video translation. Subsequently, we propose a novel approach to controllable video portrait synthesis, inspired from Implicit Neural Representations (INR). In the second part of the thesis, we focus on person-agnostic methods and present a GAN-based framework that performs video portrait reconstruction, full head reenactment, expression editing, novel pose synthesis and face frontalisation.

Acknowledgements

First of all, I am very thankful to my supervisor, Prof. Stefanos Zafeiriou. He is the one that gave me the opportunity to pursue a Ph.D. at Imperial College London and welcomed me to his lab. I would like to express my gratitude to Stefanos, for offering me an internship at FaceSoft and supervising my research work at Huawei, during the last two years of my Ph.D. All that time, Stefanos provided me with very useful guidance and helped me develop a mindset for solving research problems that matter. I am honoured to be part of the rich academic environment he fostered, where I had the pleasure to meet and work with many talented people and grow as a scientist.

I am also extremely fortunate to work with my co-supervisor Dr. Viktoriia Sharmanska. I would like to thank her for the continuous support throughout the course of my Ph.D. Viktoriia gave me the freedom to explore and pursue my own ideas. She was always eager to discuss new research directions and provide me with insightful thoughts. Many times I have left our meetings feeling inspired and motivated. Viktoriia is a very positive and supporting person and I feel very grateful to be advised by her.

This dissertation is a result of a series of collaborations with brilliant individuals. I am very grateful to all my colleagues and fellow researchers who joined their efforts with me, especially Mohammad Rami Koujan, Evangelos Ververas and Stylianos Ploumpis. Their support has been extremely valuable to me.

The department of Computing at Imperial College London has been a great place to develop research ideas. Over the past four years, I had a great time, met amazing people and formed strong friendships. I deeply thank Imperial College London for offering me a Scholarship and funding my research studies.

In closing, I want to express my gratitude to my parents and sister for their unconditional support, love and belief in me that kept me going. My family taught me to be relentless when chasing my dreams. Last but not least, I would like to thank my partner Angelina, for making the last years of my Ph.D. a pleasant journey.

Contents

Abstract vii

Acknowledgements ix

1 Introduction 1

1.1 Objectives . . . 3

1.2 Thesis Outline & Contributions . . . 4

1.3 Applications & Ethical Issues . . . 6

1.4 Publications . . . 6

2 Background & Preliminaries 8

2.1 Human Face Tracking . . . 9

2.1.1 Facial Landmark Detection . . . 9

2.1.2 3D Face Reconstruction . . . 10

2.1.3 Eye Gaze Estimation . . . 12

2.2 Image and Video Synthesis with GANs . . . 13

2.2.1 Generative Adversarial Networks . . . 13

2.2.2 GAN-based Image and Video Translation . . . 15

2.3 Video Portrait Synthesis and Reenactment . . . 21

2.3.1 Face Reenactment and Visual Dubbing . . . 21

2.3.2 Full Head Reenactment . . . 22

2.3.3 Neural Representations for Scenes and Faces . . . 23

I Person-Specific Reenactment with Complete Head Control 24

3 Head2Head: Video-based Full Head Reenactment 25

3.1 Face Tracking . . . 26
3.1.1 3D Facial Shape and 3DMMs . . . 26
3.1.2 3DMM Video Fitting . . . 27
3.1.3 Conditional Face Representation . . . 28
3.2 Video Neural Rendering . . . 29
3.2.1 Generator . . . 30
3.2.2 Discriminators . . . 30
3.2.3 Facial Optical Flow . . . 31
3.2.4 Objective functions and Optimisation . . . 31
3.3 Experimental Analysis . . . 32
3.3.1 Quantitative Results . . . 33
3.3.2 Qualitative Evaluation . . . 34
3.3.3 Ablation Study . . . 36
3.3.4 Automated Study . . . 37
3.3.5 User Study . . . 37
3.4 Multiperson Synthesis with Few-shot Learning . . . 38
3.4.1 Initial Training Stage . . . 39
3.4.2 Fine-tuning on Unseen Identities . . . 41
3.4.3 Experiments . . . 42
3.5 Summary . . . 45

4 Dynamic Neural Portraits 47

4.1 2D-coordinate MLP with Controllable Dynamics . . . 49
4.2 Dynamic Neural Portraits . . . 50
4.2.1 Audio-driven Portrait Synthesis . . . 51
4.2.2 Objective Function and Optimisation . . . 51
4.3 Experiments . . . 52

4.3.1 Implementation Details . . . 52
4.3.2 Evaluation Metrics . . . 53
4.3.3 Comparison with State-of-the-Art . . . 56
4.3.4 Ablation study . . . 60
4.4 Discussion . . . 61
4.5 Summary . . . 61

II Person-Generic Methods for Head Animation and Editing 62

5 HeadGAN: One-shot Head Synthesis and Editing 63
5.1 Conditional Inputs . . . 64
5.1.1 Face Representation . . . 64
5.1.2 Audio Features . . . 66
5.2 HeadGAN Learning Framework . . . 67
5.3 Advantages of 3D face modeling . . . 71
5.4 Experimental Evaluation . . . 72
5.4.1 Implementation details . . . 72
5.4.2 Evaluation Metrics . . . 72
5.4.3 Comparison with baselines . . . 73
5.4.4 Editing the Expression and Pose of Image Portraits . . . 77
5.4.5 Ablation Study . . . 77
5.5 Summary . . . 79

6 Reenactment with Canonical Keypoints & Explicit Gaze Control 80
6.1 Free-HeadGAN model . . . 81
6.1.1 Computation of Canonical Key-points . . . 81
6.1.2 Gaze estimation . . . 83
6.1.3 Image Synthesis . . . 85
6.1.4 Free-HeadGAN inference . . . 87
6.2 Experimental Results . . . 88

6.2.1 Dataset and Training . . . 88
6.2.2 Comparison with State-of-the-Art . . . 88
6.2.3 Ablation Study . . . 92
6.2.4 Pose and Gaze Manipulation . . . 94
6.3 Limitations . . . 95
6.4 Summary . . . 95

7 Conclusion **96**
7.1 Summary of Achievements . . . 96
7.2 Ethical Considerations . . . 98
7.3 Future Directions . . . 98

A HeadGAN Additional Results **99**

B Free-HeadGAN Additional Results **102**

Bibliography **102**

List of Tables

2.1 Checklist of key features and design choices of state-of-the-art head synthesis systems that support full head animation and reenactment 23

3.1 Numerical results of the ablation study. 36

3.2 The ratings of 95 participants on the task of self-reenactment. In columns numbered from 1 to 5 we show the number of people that gave that rating. The next column, named "real", displays the percentage of participants that rated the video with a 4 or a 5. 38

3.3 The ratings of 95 participants on the task of face reenactment and full head reenactment. 38

3.4 Numerical comparison of our proposed Head2Head extension with X2Face [WKZ18] and vid2vid [WLZ+18a] on the task of reconstruction, for five different target videos-identities. We employ the average pixel distance to measure the fidelity of reconstruction, which reveals that our method outperforms the two baselines. 43

3.5 Average pixel distance between generated and ground truth samples. We experimented with four videos-identities and tested our model with or without the initial learning stage. The reported numbers show the significance of the first training stage. 44

3.6 Quantitative results for the significance of networks D^M and D^V. 45

4.1 Details of video portraits used throughout the experiments 54

4.2 Numerical comparison with FOMM [SLT+19b], DVP [KGT+18] and NerFACE [GTZN21] for three different portraits on the task of reconstruction (self-reenactment). 56

4.3 Numerical comparison with FOMM [SLT+19b], HeadGAN [DZS21] and DVP [KGT+18] on the task of cross-identity motion transfer (reenactment). 57

4.4 Numerical comparison with AD-NeRF [GCL+21] on audio-driven reconstruction. 58

4.5 Comparison of the execution time of our generative model with various related methods. Time is reported in milliseconds (msec). We note that the results shown below refer to the forward pass time of models during inference, without considering data pre-processing such as face cropping, 3DMM fitting or audio features extraction. 59

4.6 Quantitative results of our ablation study on two separate video portraits (Biden and Obama), of resolution 512×512. 61

5.1 The architecture of dense flow network F. 68

5.2 The architecture of rendering network R. 69

5.3 Quantitative results on the tasks of reconstruction for VoxCeleb [NCZ17] test set. For all metrics, except from PSNR and CSIM, lower is better. 74

5.4 Quantitative results on the task of reenactment for VoxCeleb [NCZ17] test set. For all metrics, except from CSIM, lower is better. 75

5.5 Quantitative results on frontalisation. 77

5.6 Ablation study numerical results. Please note that due to time constraints, we trained all models for half epochs in this experiment. 78

6.1 Numerical comparison with state-of-the-art methods on the task of self-reenactment (same-identity reconstruction) for VoxCeleb [NCZ17] test set. 90

6.2 Numerical comparison with state-of-the-art methods on the task of reenactment (cross-identity motion transfer) for VoxCeleb [NCZ17] test set. 90

6.3 Evaluation of Free-HeadGAN N-shot extension, in case more than one source images are available. 92

6.4 Ablation study results on the significance of Free-HeadGAN components on the task of reenactment. 93

6.5 Comparison between the 3D mesh and 3D vector regression approaches to gaze estimation, on within-dataset, cross-subject experiments. 94

List of Figures

1.1 The hypothesised emotional response of humans against realism of a human-like entities. The uncanny valley corresponds to the region of negative emotional response towards subjects that seem "almost" human. Motion amplifies the emotional response, suggesting that photo-realistic face synthesis is even more challenging for video data. Image adapted from Wikipedia (https://en.wikipedia.org/wiki/Uncanny_valley). 2

2.1 Conditional GANs can be used to close the gap between renderings produced by computer graphics methods and real images of faces. For example, DVP learns to translate renderings of 3D faces into photo-realistic images. Image adapted from [KGT+18]. 8

2.2 Visualisation of various types of facial landmarks. Key-points may range from sparse (e.g. 5 points) to more dense sets (e.g. 68 points). In addition, they might be 2-dimensional or include depth information (3-dimensional). Here, 2D landmarks in the 3rd image were obtained using dlib library [Kin09], while all others, in the $1^{st}, 2^{nd}$ and 4th image, were extracted with RetinaFace [DGV+20a]. 9

2.3 A mesh is composed of vertices and triangles. Image adapted from [DGV+20a]. . 10

2.4 (a) pix2pix generator for image-to-image translation follows an encoder-decoder architecture with skip connections. (b) The discriminator learns to classify between synthesised images $G(x)$ and real samples y, by observing also the conditional input x. Image adapted from pix2pix [IZZE17]. 17

2.5 The architecture of pix2pixHD generator G, consists of G_1 and G_2. First, G_1 is trained on lower resolution images and then G_2 is appended, with the entire system trained on images of higher resolution. The input to the residual blocks of G_2 is the sum of the feature map from G_2 and the last feature map from G_1. Image adapted from [WLZ+18b]. 18

2.6 Different from pix2pixHD [WLZ+18b], the generator of [PLWZ19] (a) is a decoder that consists of SPADE ResNet blocks (b) and up-sampling layers. Each ResNet block contains two SPADE layers (c), which normalise their input with modulation parameters $\gamma(\mathbf{x})$ and $\beta(\mathbf{x})$ that are computed with convolutional neural networks from semantic segmentation maps (generator's conditional input $\mathbf{x}$). The image is adapted from [PLWZ19]. 19

2.7 The architecture of vid2vid generator. The conditional input $\mathbf{x}_{t-L:t-1}$ (e.g. semantic maps) is processed by a sequence of down-sampling and residual blocks. A similar encoding branch processes the past synthetic frames $\tilde{\mathbf{y}}_{t-L:t-1}$ (previous images). The two resulting feature maps are added together and passed through two identical decoding branches, one that hallucinates $\hat{\mathbf{y}}_t$ (intermediate image) and another that predicts the optical flow $\tilde{\mathbf{w}}_{t-1}$ (flow map) for warping $\tilde{\mathbf{y}}_{t-1}$ and the occlusion mask $\tilde{\mathbf{m}}_t$ (mask). Image adapted from vid2vid [WLZ+18a]. 21

3.1 Our face tracking pipeline is used to recover the camera and shape parameters of the source and target faces and track their gaze. This stage produces the conditional input for our video rendering network. During inference, we take the expressions as well as the translation and rotation parameters from the source frames. 29

3.2 The proposed rendering network is based on the architecture of the generator introduced in vid2vid [WLZ+18a]. However, unlike vid2vid, we use a single decoding branch. 30

3.3 Examples of frames from the videos we used throughout our experiments. 33

3.4 Quantitative results for the self-reenactment task, using four identities. For each target identity we show a synthesised frame example and he corresponding error map with respect to the the corresponding real frame. In the top-right corners of heatmaps we report the average reconstruction error, across the entire test sequence. 34

3.5 Cycle-reenactment results of our method. From first to last row: source frames, synthetic frames of the target subject, synthetic frames of the source identity driven by the synthetic frames of the target identity, heatmaps with per-pixel l_2-distance between the ground and generated source frames. In the top-right corners of heatmaps we present the mean per-pixel distance for each frame. . . . 34

3.6 (a) Visual comparison of the proposed Head2Head with pix2pix baseline [IZZE17], conditioned on PNCC images, for the task of self-reenactment. As can be seen, we achieve superior image quality and more accurate reconstruction. (b) Qualitative comparison of Head2Head with vid2vid [WLZ+18a] conditioned on landmark sketches. The adoption of a 3DMM in conjunction with PNCC inputs achieves identity and expression disentanglement, as our rendering network preserves the characteristics of the target, while vid2vid distorts them. 35

3.7 Comparison with DVP [KGT+18]. From top to bottom rows: source frames, target frames produced by Head2Head, the same frames generated with DVP. Our generated head poses and facial expressions follow the source performances closer than DVP. 35

3.8 Examples of frames synthesised by Head2Head, with and without the dedicated mouth discriminator D_M. It is evident that network D_M improves the quality of teeth. 37

3.9 The extended Head2Head framework in the initial training stage. The identity embedder estimates the appearance feature $\mathbf{h}^{(\mathbf{i})}$, which is afterwards injected into network G using AdaIN layers. 41

3.10 Our approach generates samples of higher quality samples when compared to X2Face [WKZ18] and vid2vid [WLZ+18a]. 43

3.11 Visual comparison of our generated samples with and without the initial learning step. 44

3.12 Mean Intersection over Union (IoU) for five different videos-identities. 45

4.1 In contrast to traditional image-to-image translation approaches for portrait rendering, such as DVP [KGT+18] and Head2Head [KDRZ20] (a), or recent NeRF-based portrait rendering methods, namely NerFACE [GTZN21] and AD-NeRF [GCL+21] (b), we propose Dynamic Neural Portraits, a novel architecture paradigm for video portrait synthesis (c). 48

4.2 Our baseline model, namely 2D-coordinate MLP with controllable dynamics (c), is based on 2D NeRF (b), a simplification to NeRF (a) after replacing 3D spatial coordinates with 2D. 49

4.3 An overview of Dynamic Neural Portraits method. 51

4.4 The architecture of our MLP network is based on the one originally proposed for NeRF [MST+20], with the addition of a linear output layer that returns the RGB colour $\mathbf{c}$ and feature vector $\mathbf{f}$. Here, γ denotes the function of positional encoding. 52

4.5 The architecture of our decoder is based on the generator of pix2pixHD [WLZ+18b], with six residual layers followed by two up-sampling layers and a final output layer. For more details on the structure of the residual block, please refer to [WLZ+18b]. 53

4.6 Illustration of the different identities (videos) we used throughout our experiments. 54

4.7 Visual comparison with the baselines on the problem of reconstruction (self-reenactment). Our method consistently generates more realistic samples with finer details than its counterparts [GTZN21, KGT+18, SLT+19b]. Note that for the correct evaluation of FOMM, images needed to be further cropped, thus appearing zoomed in the face. 57

4.8 Visual comparison with baselines on reenactment. Our approach transfers pose, expression and gaze more reliably than both person-specific DVP [KGT+18] and person-generic methods HeadGAN [DZS21] and FOMM [SLT+19b]. Please note that for the correct evaluation of HeadGAN and FOMM, images had to be cropped closer to the face. 58

4.9 Visual comparison with AD-NeRF [GCL+21] on audio-driven reconstruction. We generate more accurate lips motions and higher quality details compared to AD-NeRF. 59

4.10 Qualitative results of our ablation study. Our full model (F) (Dynamic Neural Portraits) improves the frames synthesised by our baseline (A) (2D coordinate-based MLP with controllable dynamics) by a noteworthy margin. This is clearly shown in the facial region of video portraits (eyes, teeth, etc.). 60

5.1 Our proposed HeadGAN method performs reenactment (b), by fully transferring the facial expressions and head pose from a driving frame to a reference image. When the driving and reference identities coincide (c), it can be used for facial video compression and reconstruction. In addition, HeadGAN can be applied to facial expression editing (d), novel head pose synthesis (e), and face frontalisation (f). 64

5.2 Data pre-processing stage. We recover and render the 3D face of the reference image $\mathbf{y}_{ref}$, as well as the driving frame $\mathbf{y}_t$, after adapting the identity parameters. 66

5.3 Overview of the dense flow network F. This module computes a flow field for warping the reference image and features, according to its 3D face inputs (PNCC representations). 68

5.4 Overview of the rendering network R. This network uses the visual information of the warped reference image and warped features along with the audio features in order to translate the driving PNCC input into a photo-realistic image of the reference identity. 69

5.5 Our SPADE and AdaIN blocks are based on the SPADE Resnet blocks proposed in [PLWZ19] without a residual component, as we always keep the same number of input channels F at the output, both on SPADE and AdaIN blocks. 70

5.6 Qualitative comparison of HeadGAN with four baseline methods on the task of reconstruction (self-reenactment). 74

5.7 Qualitative comparison with baselines, on the task of reenactment. FOMM-Rel refers to relative key-point coordinates strategy and FOMM-Abs to absolute key-point coordinates, used in First Order Motion Model [SLT+19b]. 75

5.8 Qualitative comparison with psp and RaR methods on face frontalisation. 76

5.9 Expression and pose editing with HeadGAN. Please note that we use our model's variation without audio input (AdaIN blocks) for this task. 77

5.10 The significance of dense flow network F in the quality of images. 78

5.11 The identity preservation problem becomes prominent when conditioning image synthesis on sketches of facial landmarks, instead of PNCC images. 78

6.1 The pipeline for computing the canonical key-points during training. Using a pair of frames coming from the same video, we train E_{can} to disentangle identity from expression and pose. We project the regressed 3D key-points into the canonical space and then try to reconstruct them, after swapping the canonical points of the source and target images. 82

6.2 Outline of our gaze estimation method. Having defined an eyeball template, $\mathbf{T}_{eye}$ we generate pseudo-ground truth meshes for available gaze estimation datasets and employ them to train our 3D eye mesh regression network E_{gaze}, which given input images produces 3D coordinates adhering to $\mathbf{T}_{eye}$. 84

6.3 Overview of the image generation component. The generator is made up from two modules: a flow network and a rendering network. The flow network computes the optical flow for warping the reference image and features, according to the target sketch. The rendering network uses this visual information, in order to translate the driver's sketch into a photo-realistic image of the source. The two networks are trained jointly in an adversarial manner, with the assistance of image discriminators. 85

6.4 Illustration of optical flows and weights predicted by the flow network, in case four source images are provided instead of one (4-shot). The warped image is calculated from the source images, flows and weighs with the application of Eq. (6.12). 86

6.5 End-to-end pipeline of Free-HeadGAN during inference, for the task of reenactment. We adapt the target key-points to the facial shape (identity) of the source. 87

6.6 We evaluate the generative performance of Free-HeadGAN on the tasks of (a) same-identity reconstruction (self-reenactment) and (b) cross-identity motion transfer (reenactment). 89

6.7 Visual comparison of our method with baselines on the task of self-reenactment. 91

6.8 Qualitative evaluation of our method against baselines on the task of reenactment. 91

6.9 Synthetic examples when using one, two, four or eight source images in the N-shot scenario. The results improve in terms of image quality as we get access to more source inputs. 93

6.10 Examples where explicit gaze control is essential for successful gaze transfer, as 68 facial landmarks appear to be insufficient. 94

6.11 The 3D key-point modeling of faces enables our system to edit the head pose of a given reference image. Furthermore, we have explicit gaze control over the subject. 95

A.1 Reconstruction examples with HeadGAN. From left to right: reference, reference PNCC image, driving PNCC image, flow, warped, generated, driving. 100

A.2 Reenactment examples with HeadGAN. From left to right: reference, reference PNCC image, driving PNCC image, flow, warped, generated, driving. 101

B.1 Reenactment results with Free-HeadGAN (a) 102

B.2 Reenactment results with Free-HeadGAN (b) 103

B.3 Reenactment results with Free-HeadGAN (c) 103
B.4 Reenactment results with Free-HeadGAN (d) 104
B.5 Reenactment results with Free-HeadGAN (e) 104
B.6 Reenactment results with Free-HeadGAN (f) 105

Chapter 1

Introduction

"A picture is worth a thousand words" is a commonly used adage, which means that a great deal of information can be enclosed within a single image and communicated more effectively than a verbal explanation. In the era of social media, people share their feelings, emotions, thoughts and ideas by capturing and sharing pictures and videos (e.g. selfies, short videos). It is evident that when it comes to digital communication and online interactions, audio-visual data that portray human faces are more important than ever. The abundance of such data on the internet has led to the development of systems capable of analysing the actions of faces from images and videos in an automated way. These systems find applications in various problems, including face recognition and tracking, identity recognition, expression and emotion recognition, as well as 3D face reconstruction, with the solutions being given mainly using techniques from *Computer Vision (CV)* and *Computer Graphics (CG)*.

With the rise of *Deep Learning (DL)*, Computer Vision was among the fields of *Artificial Intelligence (AI)* that got revolutionised, as *neural networks* were proven very effective when it comes to learning complicated patterns from vast amounts of visual data. As a consequence, important advancements have been made in designing methods that learn to examine human behaviour from image and video data. Nonetheless, apart from utilising neural networks for tracking and analysing the appearance and movements of human faces, recent advances in *Deep Generative Models (DGMs)* have shown very promising results towards the direction of synthesising photo-realistic images or videos of faces.

Generating realistic visual data of human faces is an extremely challenging task. This is mainly attributed to the *uncanny valley* phenomenon, an unsettling feeling people experience when audio or visual data that resemble human performance appear convincing in some aspects, but not quite realistic. As hypothesised, the negative emotional response is amplified in case the artificial entity is moving, which suggests that creating photo-realistic videos of faces is an even more laborious task when compared to image synthesis. This concept is illustrated in Fig. 1.1.

Arguably, the success of recent methods for realistic visual data synthesis is largely attributed to the development of *Generative Adversarial Networks (GANs)* [GPAM+14] and their capability to produce samples that on many occasions seem indistinguishable from real data. GANs learn to generate data from a distribution that mimics the training data distribution. For instance, a GAN trained on a database of photographs with human faces (e.g. image portraits) can produce photos of novel identities that do not exist in the training set, but appear authentic to human observers. In a nutshell, a GAN framework consists of two neural networks, a

generator and a discriminator, competing with each other in a zero-sum game. The generative network generates data samples, while the discriminative network evaluates their authenticity. The seminal GAN architecture was introduced as an *unconditional* generative model for unsupervised learning [GPAM+14]. In this framework, the generator network, parameterised by a *Multilayer Perceptron (MLP)*, learns to map noise vectors randomly drawn from a latent space to the data distribution of interest. Following the adversarial objective of GANs, a more constrained class of networks was designed based on *Convolutional Neural Networks (CNNs)*, namely *Deep Convolutional Generative Adversarial Networks (DCGANs)* [RMC15], which led to more stable image synthesis. Since then, numerous GAN-based approaches to unsupervised unconditional image generation have been proposed [KALL17, KLA19, KLA+20], leading to increasingly better visual results. At the same time, various methods based on *conditional GANs* [MO14] have sought to solve the problem of *image-to-image translation* [IZZE17, WLZ+18b]. These models paved the way for multi-domain image translation [CCK+18, CUYH20], which have shown impressive results when it comes to editing facial attributes of image portraits.

Unlike images, synthesising photo-realistic videos of faces, also known as *video portraits*, is a less studied problem. Until recently, the task of generating or editing video portraits was predominantly approached by the computer graphics community, with various attempts focusing on the problem of *face reenactment* [GVR+14, TZN+15, TZS+16, TZT+18]. This involves transferring the facial deformations of a source actor to another subject, typically by re-writing the inner face pixels of the target video with the assistance of 3D modeling. In order to control both the facial expressions and head pose of the produced video, [KGT+18] introduced a *person-specific* deep generative network that performs *full head reenactment*, by translating physical-based renderings of a parametric 3D face model into photo-realistic frames of the target

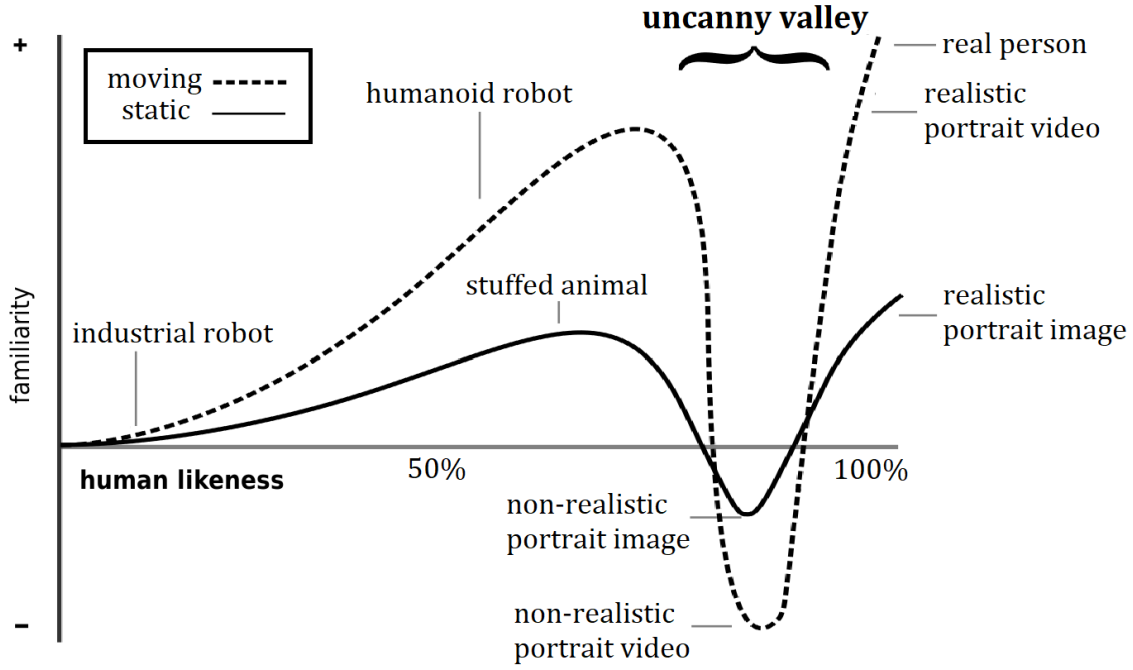


Figure 1.1: The hypothesised emotional response of humans against realism of a human-like entities. The uncanny valley corresponds to the region of negative emotional response towards subjects that seem "almost" human. Motion amplifies the emotional response, suggesting that photo-realistic face synthesis is even more challenging for video data. Image adapted from Wikipedia (https://en.wikipedia.org/wiki/Uncanny_valley).

actor. This formulation requires optimising a separate network for each target identity, using a long sequence of training frames. In case where a few training frames are available, optimising a *person-generic* generative network with adaptive normalisation layers using a few-shot learning strategy has been recently proven to be an effective approach [ZSBL19, ZISL20].

1.1 Objectives

How can we build systems that are able to generate photo-realistic visual data of human faces, while explicitly controlling attributes such as head pose, facial expression and eye gaze? In this thesis, we mainly focus on the problem of reenactment, as we aim to develop novel methods for synthesising image and video portraits, by transferring motion information from a source actor to a target identity. Motivated by the success of graphics-based methods in face reenactment, as well as the recent advances in deep generative modeling, we take a step towards harnessing the power of neural networks to address the problem of controllable video portrait synthesis.

In order to approach such a problem, one needs to deal with two separate issues, namely: (i) how to obtain accurate information related to head movements, such as pose, expression, gaze from visual data, and (ii) how to utilise this information to generate realistic video portraits that closely follow the driving performance without compromising the target's identity. Considering the above, the goal of this work is to develop generative approaches that combine concepts from deep learning (i.e. GANs) and 3D face modeling, such as *3D Morphable Models (3DMMs)* [BV99, BRP+18]. This coupling allows our proposed methods to enjoy the benefits of both worlds, as they rely on very precise representations of faces and follow the most recent advances in deep learning and neural networks. In the course of this thesis, we explored both identity-generic and identity-specific approaches, inspired by concepts from video modeling, video-to-video translation, optical flow prediction and implicit neural representations. In particular, throughout this research we pursued the following directions:

- Over the last years, deep learning methods have been successfully applied to photo-realistic image synthesis. More recently, image-to-image translation networks were proposed for generating controllable video portraits frame-by-frame. Nonetheless, the majority of existing solutions to person-specific full head reenactment are image-based, as they do not explicitly model the temporal dependence between generated frames. As a result, the synthesised sequence of frames might exhibit temporal inconsistencies. Is it possible to apply ideas from video-to-video translation in order to improve the quality of reenactment, in terms of temporal stability and fidelity of expression transfer?

- Most of the existing deep learning architectures for full head reenactment perform image translation using purely CNN-based neural rendering networks in an encoder-decoder formation. This design choice requires the preparation of 2D inputs for the neural networks, such as 2D facial sketches drawn from landmarks or renderings of 3D faces. Would it be possible to disengage our generative model from these 2D facial representations, with a new paradigm for portrait rendering? Could a hybrid MLP and CNN architecture yield higher inference speeds and superior visual quality?

- Recently, a significant amount of research has been focused on reenacting human heads using a limited amount of data and relying on few-shot learning strategies, even when a

single reference image of the target person is available (one-shot). The vast majority of previous attempts represents faces with the assistance of sparse landmarks (facial key-points). This leads to severe identity preservation issues during cross-identity motion transfer, as identity-related information of the driving person encoded within landmarks is passed to the generated frames. Could we solve this problem by representing faces with a parametric 3D model that disentangles expression-related deformations from facial shapes? To what extend can we benefit by warping the reference image using a dense optical flow that is estimated with the assistance of a semantic facial representation?

- Could we employ an alternative to 3DMMs and parametric 3D face modeling for disentangling identity from expression information? How can we combine a novel architecture for one-shot full head reenactment with the concept of canonical key-points to achieve identity preservation? How can we accurately track eye gaze and utilise it to drive synthesis?

1.2 Thesis Outline & Contributions

This thesis is organized into two parts. The first part is about person-specific approaches to the problem of controllable video portrait synthesis and reenactment. The second part focuses on person-agnostic models for face animation and editing using one-shot learning. In the following, we provide an outline of the contributions presented in each chapter of this thesis.

Part I: Person-Specific Reenactment with Complete Head Control

In this part, we propose two different approaches to the problem of full head reenactment, which includes transferring the facial expression, head pose and eye gaze from any source actor to a specific target identity. This first part consists of two chapters:

Chapter 3: Head2Head: Video-based Full Head Reenactment

In this chapter, we present *Head2Head* [KDRZ20], a video-based approach to controllable video portrait synthesis. In Section 3.1, we start by presenting our face tracking pipeline. In Section 3.2, we continue by introducing our deep generative model. In contrast to previous image-based attempts that generate video frames independently from each other, we draw inspiration from recent progress in video-to-video translation [WLZ+18a] and propose a conditional GAN with a sequential generative network and a temporal discriminator. We condition synthesis on PNCC images [ZLL+16], a rendering of 3D faces that encapsulates semantic information of the facial region. During training, we include an additional discriminator network to our GAN framework, which is dedicated to the mouth area and forces the generator to produce samples with finer teeth details, a region where no conditional information is available. Our experiments in Section 3.3 demonstrate the significance of our design choices and the merits of video modeling. Then, in Section 3.4, we present an extension to Head2Head in the direction of few-shot learning, with the incorporation of an identity encoder network. We describe how to train our extended framework on a dataset of multiple faces, and then how to fine-tune its parameters using a short unseen video of an new target identity.

Chapter 4: Dynamic Neural Portraits

In Chapter 4 we present *Dynamic Neural Portraits*, a novel paradigm to controllable video portrait synthesis. Our approach is completely different from existing reenactment systems

that rely on GANs and image-to-image translation networks for transforming renderings of 3D faces into photo-realistic images. Instead, we build our system upon a 2D coordinate-based MLP with controllable dynamics, which is presented in Section 4.1. Next, in Section 4.2, we show how to combine this network with a CNN-based decoder and formulate our complete method. Primarily, we condition our model on expression blendshapes, head pose and eye gaze parameters. Nonetheless, we show that apart from expression parameters, our architecture can be successfully driven by audio features. Our qualitative and quantitative experiments in Section 4.3 demonstrate that the proposed architecture provides a significant speedup in comparison with recent state-of-the-art NeRF-based methods for video portraits synthesis [GTZN21, GCL+21], while outperforming prior works in terms of visual quality.

Part II: Person-Generic Methods for Head Animation and Editing

While identity-specific methods generate high quality results with convincing head movements and facial dynamics, they require thousands of training frames for each target person and train a separate model per subject. In the second part of this thesis, we move towards identity-generic models, which require just a single (or a few) reference image(s) of the target identity.

Chapter 5: HeadGAN: One-shot Head Synthesis and Editing

This chapter introduces *HeadGAN* [DZS21], a neural architecture for one-shot video portrait synthesis. Given a reference image portrait from one person and a driving sequence of frames from another person, our method performs full head reenactment while preserving the target identity. As we describe in Section 5.1, our system conditions synthesis on renderings of 3D faces (PNCCs). These renderings are extracted from the driving video and then adapted to the facial geometry of a reference image, in order to disentangle identity from expression. We also improve the mouth movements, by optionally using audio-related features as a second input. In Section 5.2, we demonstrate the novel architecture of HeadGAN, which is made up of a flow network for the prediction of dense optical flow and a neural rendering network. Both modules are equipped with spatially-adaptive normalization layers [PLWZ19]. In Section 5.4, we perform extensive experiments, which showcase that when it comes to one-shot reenactment our method performs better than many state-of-the-art systems for talking head synthesis and motion transfer by a noteworthy margin. Lastly, we show that the adopted conditional face representation allows HeadGAN to be utilised as an efficient method for video portrait compression and reconstruction, as well as a powerful tool for editing the facial expression and head pose of image portraits.

Chapter 6: Reenactment with Canonical key-points & Explicit Gaze Control

In this chapter, we describe *Free-HeadGAN*, a variation of HeadGAN [DZS21] that is released from the parametric 3D models of the face. Instead of relying on past knowledge of identity and expression disentanglement captured by 3DMMs, we show that modeling faces with sparse canonical 3D landmarks are sufficient for achieving comparable generative performance with HeadGAN. In Section 6.1, we describe our complete pipeline, which consists of three components: a canonical 3D key-point estimator that recovers 3D pose and expression-related deformations, a gaze estimation network and an image generator. Apart from 3D pose and facial expressions, our method is also capable of transferring the eye gaze. In Section 6.2, we experiment with an extension of our one-shot generator to accommodate few-shot learning using an attention mechanism, in case more than one reference images are available. Finally, we present a detailed evaluation of the proposed system.

1.3 Applications & Ethical Issues

Without a doubt, the applications of reenactment systems are numerous. For instance, they can be used to efficiently compress long video portraits into parameters of pose, expression and gaze, and then reconstruct the original footage with the assistance neural rendering networks. One-shot identity-generic methods can serve as tools for editing the facial expressions and head pose of image portraits. In addition, transferring the complete head movements of a source actor to a target identity could find applications in the movie industry, for video editing and movie dubbing. In social media, reenactment methods could enable content creators produce entertaining videos. Furthermore, video portrait synthesis could enable people to enjoy more privacy in teleconferences, by utilising a photo-realistic avatar of themselves. Other possible applications include virtual assistants, virtual and augmented reality and video games.

Despite their importance in many applications, synthetic visual data could be potentially used for unethical purposes, known as "deepfakes". For instance, generative methods may be employed to create fake videos of well-known people uttering things they would never say. Therefore, it becomes obvious that the ethical implications and the potential harmful applications of such a technology should be openly discussed. As the authors of [DR21] pointed out, there are three relevant factors for determining whether a synthetic image or video is morally controversial: (i) whether the portrayed persons would have an objection regarding the way in which they are represented, (ii) whether the visual data deceive the viewers, and (iii) the intent with which the creator generated the footage.

Considering that synthetic visual data are becoming more and more realistic and might appear plausible to a large number of people, we have reached an era where such concerns need to be taken seriously. For starters, it is of paramount importance that the person being displayed in synthetic audio-visual data is aware and agrees to be presented in this specific way. Therefore, an individual or organisation that generates synthetic media needs to obtain the consent of each portrayed person, prior to synthesising and releasing the footage. Additionally, the generated material should not produce fake news or controversial opinions that misguide the audience to form a wrong perception or view about significant matters. To that end, the intentions behind AI-generated data are a key factor. Similarly to many other technological achievements, synthetic media may be used for a harmful or benign purpose. For this reason, it is essential to acknowledge the possible risks involved within controllable video portrait synthesis, promote only the morally valuable applications and denounce any potential misuses.

1.4 Publications

The content of this thesis is composed of material from the following publications:

- Mohammad Rami Koujan*, Michail Christos Doukas*, Anastasios Roussos, Stefanos Zafeiriou, **Head2Head: Video-based neural head synthesis**, In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020).*
- Michail Christos Doukas*, Mohammad Rami Koujan*, Viktoriia Sharmanska, Anastasios Roussos, Stefanos Zafeiriou, **Head2Head++: Deep Facial Attributes Re-Targeting**, In *IEEE Transactions on Biometrics, Behavior, and Identity Science (TBIOM), 2021.*

- Mohammad Rami Koujan*, Michail Christos Doukas*, Viktoriia Sharmanska, Anastasios Roussos, Stefanos Zafeiriou, **ReenactNet: Real-time full head reenactment**, In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020).*

- Michail Christos Doukas, Mohammad Rami Koujan, Viktoriia Sharmanska, Stefanos Zafeiriou, **Head2HeadFS: Video-based Head Reenactment with Few-shot Learning**, *arXiv preprint.*

- Michail Christos Doukas, Stefanos Zafeiriou, Viktoriia Sharmanska, **HeadGAN: One-Shot Neural Head Synthesis and Editing**, In *Proceedings of the IEEE/CVF International Conference on Computer Vision 2021 (ICCV 2021).*

- Michail Christos Doukas, Evangelos Ververas, Viktoriia Sharmanska, Stefanos Zafeiriou, **Free-HeadGAN: Neural Talking Head Synthesis with Explicit Gaze Control**, *In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2023.*

- Michail Christos Doukas, Stylianos Ploumpis, Stefanos Zafeiriou, **Dynamic Neural Portraits**, In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2023 (WACV 2023).*

* equal contribution

Chapter 2

Background & Preliminaries

Creating realistic digital faces is a challenging problem, with solutions mainly given by the computer graphics community, via physically modeling human faces and using concepts from light transport to render realistic facial images. With the advent of deep generative modeling, machine learning has been proven a promising approach for tackling this problem from a data-driven perspective, by seeking to learn the distributions of real image and video data.

Deep generative models learned from large-scale data collections have show impressive results in photo-realistic synthesis of images and videos. More specifically, Generative Adversarial Networks (GANs) [GPAM+14] are established as a common approach for generating random high-resolution images that are almost identical with real data, by resembling the statistics of samples in the training set. When it comes to face animation and reenactment, being able control the generated samples is very important. Conditional GANs (cGANs) [MO14] enable direct control of the synthesised data, with the addition of a control signal to the generative algorithm. Regarding video portrait synthesis, this input signal is usually a representation of the face that guides the generative process. It may range from embeddings that encode facial motions to 2D images, such as renderings of 3D faces, created by physical-based renderers. For instance, Deep Video Portraits (DVP) [KGT+18] employ a cGAN that learns to translate renderings of 3D faces into photo-realistic images, as it adds high-frequency details on faces and completes the frame by generating the background, torso and hair (please see Fig. 2.1).



Input Output Input Output

Figure 2.1: Conditional GANs can be used to close the gap between renderings produced by computer graphics methods and real images of faces. For example, DVP learns to translate renderings of 3D faces into photo-realistic images. Image adapted from [KGT+18].

On the subject of reenactment and talking head synthesis, the conditional face representation that serves as input to cGANs has an critical impact on the design choices and performance of deep generative models, as it plays a crucial role in the architecture of neural networks. In most cases, this conditional face representation is extracted from image or video data with the assistance of pre-existing off-the-shelf face tracking systems.

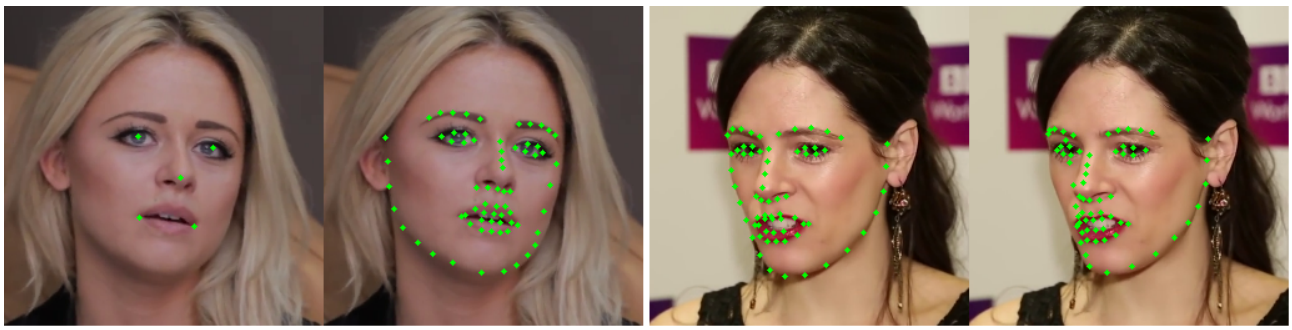
This chapter is organised in two sections. In Section 2.1, we provide an overview of widely-used methods that track human faces and recover useful information for driving face synthesis. Then, in Section 2.2, we introduce basic concepts concerning data synthesis with GANs and delve deeper into conditional image and video generation with cGANs. Finally, In Section 2.3, we give an overview of previous works related to video portrait synthesis and reenactment.

2.1 Human Face Tracking

Face tracking is most commonly associated with the problem of face detection, which comes down to identifying the presence of human faces in video frames and computing bounding boxes or critical facial landmarks. Here, we use the term of face tracking in a more general sense, which involves extracting meaningful information from human faces, such as facial shape and expression, head pose and eye gaze. Recovering and analysing head geometry and facial deformations from image or video data is a hard problem by itself. It is closely connected to face generation, as it provides generative algorithms with the essential information for guiding the process of synthesis. In this section, we give a brief overview of well-established face tracking tasks associated with our work, including landmark detection and 3D face reconstruction. Then, we discuss eye gaze detection, which is usually treated as a separate problem.

2.1.1 Facial Landmark Detection

Landmark localisation is a computer vision task that involves the detection and tracking of key-points from a human face. Given an image with a face, a landmark detector learns to predict a



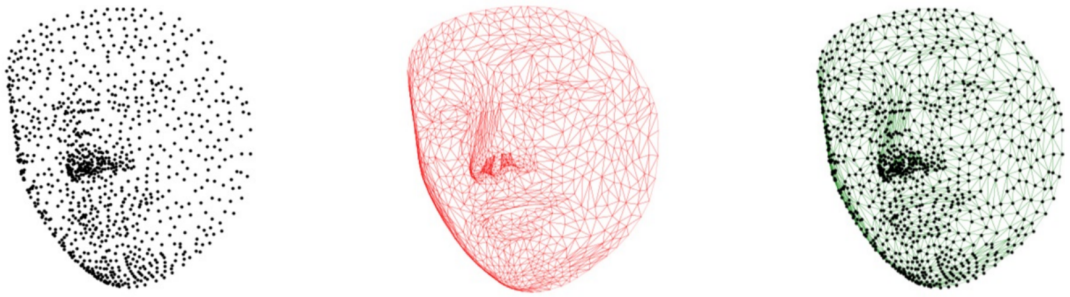
5 landmarks 68 landmarks 2D landmarks 3D landmarks

Figure 2.2: Visualisation of various types of facial landmarks. Key-points may range from sparse (e.g. 5 points) to more dense sets (e.g. 68 points). In addition, they might be 2-dimensional or include depth information (3-dimensional). Here, 2D landmarks in the 3rd image were obtained using dlib library [Kin09], while all others, in the $1^{st}, 2^{nd}$ and 4th image, were extracted with RetinaFace [DGV+20a].

set of K key-points, where each point is represented either by its 2D coordinates (x, y) on the image plain, or by 3D coordinates (x, y, z), in case depth information is available. The most common sets of landmarks are sparse (e.g. 5 or 68 key-points) and placed on important regions such as the mouth, eyes, eyebrows, nose and face contour. Figure 2.2 shows an illustration of 5 and 68 key-points, as well as the differences between 2D and 3D landmarks. As can be seen, 3D landmarks are consistent with the 3D geometry of the face. It is a common practice to train landmark detectors [FKA+18] in a supervised way using large image databases of faces which have been manually or semi-automatically annotated with semantically meaningful key-points that follow a specific enumeration [KWRB11, BAPD13, STZP13, WQY+18]. Due to their robustness on capturing rigid and non-rigid facial deformations, landmark detection algorithms find application in several problems, such as face alignment, head pose estimation, face animation and reenactment, face swapping and blinking detection.

2.1.2 3D Face Reconstruction

Apart from sparse key-points, the geometry of faces is very often represented by a number of vertices in the Cartesian space. In general, a 3D facial mesh consists of N vertexes, which can be written as a $3N$-dimensional vector $\mathbf{s} = [\mathbf{v}_1^\top, \dots, \mathbf{v}_N^\top]^\top = [x_1, y_1, z_1, \dots, x_N, y_N, z_N]^\top$, where $\mathbf{v}_i = [x_i, y_i, z_i]^\top$ are the 3D coordinates of vertex i. These vertices have a fixed topology with semantic ordering. In most cases, vertices are coupled with a set of triangles (triangulation), which define a 3D mesh, as depicted in Fig. 2.3.



3D vertices triangles 3D mesh

Figure 2.3: A mesh is composed of vertices and triangles. Image adapted from [DGV+20a].

3D Morphable Face Models Since their introduction, 3D Morphable Models (3DMMs) have been widely used as a general face representation [BV99, PKA+09, BRP+18, CWZ+13, LBB+17, EST+20]. 3DMMs are strong statistical models learned from data collections of 3D facial meshes, with M shapes $\{\mathbf{s}_j\}_{j=1}^M$. In the seminal work [BV99], each shape $\mathbf{s}_j$ is accompanied by a texture vector $\mathbf{t}_j = [R_1, G_1, B_1, \dots, R_N, G_N, B_N]^\top$ that contains the RGB color values of the N corresponding vertices. Assuming that exemplar meshes are in full correspondence, the key idea of 3DMMs is to separate the shape of the face from texture, but also disentangle these from other factors, like camera position and illumination. A *shape model* can be constructed by first bringing the corresponded training meshes $\{\mathbf{s}_j\}$ into a shape space after applying Generalized Procrustes Analysis (GPA). This is followed by a Principal Component Analysis (PCA), which yields a mean shape vector $\bar{\mathbf{s}} \in \mathbb{R}^{3N}$ and an orthonormal basis $\mathbf{U}_s \in \mathbb{R}^{3N \times n_s}$, after keeping the first n_s principal components. In the same way, we can build a *texture model* by running PCA on the texture data $\{\mathbf{t}_j\}$. This results in a mean texture vector $\bar{\mathbf{t}} \in \mathbb{R}^{3N}$ and

an orthonormal basis $\mathbf{U}_t \in \mathbb{R}^{3N \times n_t}$, with the first principal components of texture. Now, the shape and texture model can be employed to approximate a new test shape $\mathbf{s}_0$ and test texture $\mathbf{t}_0$ as follows:

$$\mathbf{s}_0 \approx \mathcal{S}(\mathbf{p}_s) = \bar{\mathbf{s}} + \mathbf{U}_s \mathbf{p}_s, \quad \mathbf{t}_0 \approx \mathcal{T}(\mathbf{p}_t) = \bar{\mathbf{t}} + \mathbf{U}_t \mathbf{p}_t, \tag{2.1}$$

where $\mathbf{p}_s \in \mathbb{R}^{n_s}$ are the *shape parameters* and $\mathbf{p}_t \in \mathbb{R}^{n_t}$ are the *texture parameters* of $\mathbf{s}_0$ and $\mathbf{t}_0$ accordingly. The shape model that represents the geometry of faces can be further divided into an *identity shape model* $\{\bar{\mathbf{s}}_{id}, \mathbf{U}_{id}\}$ and an *expression shape model* $\{\bar{\mathbf{s}}_{exp}, \mathbf{U}_{exp}\}$. On the one hand, an identity shape model is learned from facial meshes with anatomical variations across different individuals, based on the assumption that faces are in the neutral expression. For instance, the largest-scale 3DMM of facial identity, namely Largest Scale Facial Model (LSFM) [BRP+18], was built from around 10K scans of distinct faces, which contain information from a wide variety of the human faces, while considering gender, age and ethnicity variations. On the other hand, the expression shape model is learned from various expressions, as deviations from a reference shape $\mathbf{s}_{id}$. The individual-specific expression blendshapes of FaceWarehouse [CWZ+13] is such an expression shape model. Combining the two 3DMMs described above, we can devise a linear model that approximates a unseen 3D facial shape $\mathbf{s}_0$ as

$$\mathbf{s}_0 \approx \mathcal{S}(\mathbf{p}_{id}, \mathbf{p}_{exp}) = \bar{\mathbf{s}} + \mathbf{U}_{id}\mathbf{p}_{id} + \mathbf{U}_{exp}\mathbf{p}_{exp}. \tag{2.2}$$

Here, $\bar{\mathbf{s}} = \bar{\mathbf{s}}_{id} + \bar{\mathbf{s}}_{exp}$ is the mean shape, $\mathbf{p}_{id} \in \mathbb{R}^{n_{id}}$ are the *identity shape parameters* and $\mathbf{p}_{exp} \in \mathbb{R}^{n_{exp}}$ are the *expression shape parameters*.

3DMM Fitting Given a facial image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, we can recover shape $\mathbf{p}_{id}, \mathbf{p}_{exp}$ and texture $\mathbf{p}_t$ parameters with 3DMM fitting. This process involves defining a camera model, which aims to project the vertices of a 3D mesh into an image plane. Let's assume that this projection is carried out by a camera model $\mathcal{P}(\mathbf{s}, \mathbf{p_c}) : \mathbb{R}^{3N+n_c} \rightarrow \mathbb{R}^{2N}$ that takes a 3D shape $\mathbf{s}$ along with a vector of *camera parameters* $\mathbf{p_c} \in \mathbb{R}^{n_c}$ and returns the 2D spatial locations where each vertex is mapped. A widely-used formulation of camera parameters consists of isotropic scale $s \in \mathbb{R}$, a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ (or Euler angles) and a translation vector $\mathbf{t} \in \mathbb{R}^3$. The 3DMM image fitting is usually performed by finding the solution of a non-linear optimization problem

$$\min_{\mathbf{p}} \mathcal{E}(\mathbf{p}) = ||\mathbf{I}(\mathcal{P}(\mathcal{S}(\mathbf{p}_{id}, \mathbf{p}_{exp}), \mathbf{p_c})) - \mathcal{T}(\mathbf{p}_t)||_2^2 + \lambda_{id}||\mathbf{p}_{id}||_2^2 + \lambda_{exp}||\mathbf{p}_{exp}||_2^2 + \lambda_t||\mathbf{p}_t||_2^2 \tag{2.3}$$

and recovering the parameters $\mathbf{p} = \{\mathbf{p}_{id}, \mathbf{p}_{exp}, \mathbf{p}_t, \mathbf{p_c}\}$. Here, $\lambda_{id}, \lambda_{exp}, \lambda_t$ are hyperparameters that control the amount of regularisation on shape and texture parameters. This optimisation problem is most often solved by Gauss-Newton method. Then, the estimated optimal parameters can be used in order to compute the 3D shape of the face.

The linear model described above is not very effective in representing the structure of facial textures that appear "in-the-wild". Therefore, 3DMM fitting usually fails with "in-the-wild" images. For that, variations of this method have been proposed [BAP+17], which extend 3DMM fitting on handcrafted image features, such as Histogram of Oriented Gradients (HoGs) for representing facial texture instead of RGB colors.

3DMM Fitting based on key-points Recent advancements in the field of 3D face reconstruction have made possible to regress dense sets of 3D facial key-points from RGB images using neural networks. For instance, RetinaFace [DGV+20a] is able to predict $K = 1103$ points for each face appearing in a given input image. These key-points share the same topology with LSFM 3DMM [BRP+18], as they are a subset of the 53 thousand vertices defined in LSFM.

Given an image $\mathbf{I}$ and the predicted dense key-points $\mathbf{s}_I \in \mathbb{R}^{3K}$, we can fit a 3DMM shape model to the image *without considering texture information*, based solely on key-points $\mathbf{s}_I$. To that end, first we apply Generalized Procrustes Analysis (GPA) on $\mathbf{s}_I$ to align it with the mean shape $\bar{\mathbf{s}}$ and then solve the following minimisation problem

$$\min_{\mathbf{p}} \mathcal{E}'(\mathbf{p}) = ||\mathcal{S}(\mathbf{p}_{id}, \mathbf{p}_{exp}) - \mathbf{s}_I||_2^2, \tag{2.4}$$

which yields parameters $\mathbf{p} = \{\mathbf{p}_{id}, \mathbf{p}_{exp}\}$. Please note that camera parameters $\mathbf{p}_c$ can be retrieved as a by-product of GPA. Moreover, the shape $\mathcal{S}(\mathbf{p}_{id}, \mathbf{p}_{exp})$ given by Eq. (2.2) is computed using a subset of the 3DMM, after keeping the columns of matrices $\mathbf{U}_{id}$ and $\mathbf{U}_{exp}$ that correspond to the available K key-points, out of all N vertices. Since this method does not reconstruct facial texture, it is useful for recovering mainly shape information from "in-the-wild" images, as well as head pose in the form of camera parameters.

3D Face Reconstruction with Neural Networks The linear basis of 3DMMs is remarkable in representing the basic characteristics of the reconstructed face. Nonetheless it fails in reconstructing high frequency details of facial geometry. The problem is much more evident when modelling the texture, that is undoubtedly a significant part of face reconstruction. To that end, a significant amount of research has been focused on employing deep neural networks for 3D face reconstruction. GANFIT [GPKZ19] is one of the first methods to extend the 3DMM fitting strategy using neural networks. More specifically, it trains a GAN [GPAM+14] on a collection of high-resolution UV-maps that replaces the PCA-based texture model. On top of that, is uses a pre-trained network on the task of face recognition, in order to preserve the identity-related characteristics of the reconstructed texture and shape. DECA [FFBB21] is a more recent network-based method that learns to estimate parameters for reconstructing facial shapes from images. It relies on FLAME [LBB+17] as a shape decoder and uses a differentiable renderer coupled with a photometric loss during training. Unlike RetinaFace [DGV+20a], DECA does not require paired data for 3D shape supervision. RingNet [SBFB19] follows a similar approach, as it learns to predict FLAME parameters from single images in an unsupervised manner.

2.1.3 Eye Gaze Estimation

In general, eyes move independently of the other parts of the face, thus eye tracking and modeling is most often studied as a separate task. Typically, eye gaze detection methods predict gaze either as a 3D vector or spherical coordinates, indicating the direction a person is looking at [KRS+19, ZSFB15]. The most common approach to gaze estimation involves learning a direct mapping between image patches around the eyes (or the entire face) and gaze vectors or gaze angles, using a neural network. Ever since [ZSFB15] employed CNNs to tackle the task, appearance-based methods have been the default approach. Multiple methods attempt to recover geometric features of eyes and use them to infer gaze [PZBH18, PSH18, WZJ18]. Others focus on adapting generic gaze estimation networks to specific test domains, aiming to produce person-specific models [PMM+19, HPV+19, YLO19] or adapting to unseen image domains [GYZ+20, LLWL21]. Apart from models trained in a supervised setting using video datasets with gaze annotations, some of the latest works have shown that learning eye gaze from images is possible even with a unsupervised, or weakly-supervised algorithm [YO20, SZSC21, KDMI+21].

2.2 Image and Video Synthesis with GANs

With regards to image synthesis, the most prominent approaches are autoregressive models (e.g. PixelCNN) [VdOKE+16, VOKK16, SKCK17], variational autoencoders (VAE) [KW13] and generative adversarial networks (GANs) [GPAM+14]. Despite their merits, autoregressive models and variational autoencoders fall behind GANs in terms of visual quality and resolution, which have shown tremendous advancements recently. In this section, we make a introduction to GANs and then present an overview of its well-known applications for conditional image generation, image-to-image and video-to-video translation, which are closely related to our work. Although these models are not specifically designed for producing controllable image or video portraits, understanding their functionality is the first step towards building learning-based reenactment systems.

2.2.1 Generative Adversarial Networks

In the general GAN setting [GPAM+14], we train two networks: a generator G and a discriminator D, also known as a critic. Given a latent variable, the generator produces a sample, e.g. an image. Its objective is to learn a distribution of samples that is indistinguishable from the training data distribution p_{data}. For example, a GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers, having many realistic characteristics. The generative model is pitted against its adversary, the discriminator, which learns to detect if a sample comes from the model distribution or the distribution of data. The two networks are optimised jointly in an adversarial manner, as the generator tries to synthesise samples similar to the ones in the training set and the discriminator tries to distinguish between synthetic and real data. To learn the model distribution, denoted as p_g, first we choose a prior distribution on latent codes $p_{\mathbf{z}}(\mathbf{z}), \mathbf{z} \in \mathbb{R}^{n_{\mathbf{z}}}$. This is usually the standard normal distribution, thus $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, we define a mapping from noise to data $G(\mathbf{z}; \theta_G)$, modeled by G. This is a function represented by a neural network with parameters θ_G. We also define a differentiable critic function $D(\mathbf{x}; \theta_D)$, parameterised by another neural network. The discriminator receives a data sample and outputs a scalar value. In the original GAN framework [GPAM+14], the scalar $D(\mathbf{x}) \in [0, 1]$ denotes the probability that $\mathbf{x}$ is drawn from the actual data, rather than p_g. We train D to maximise the probability of predicting the label of both training data and synthetic samples coming from G:

$$\max_{\theta_D} \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{2.5}$$

On the contrary, network G is optimised to minimise the probability that D assigns the correct label to its synthesised samples:

$$\min_{\theta_G} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{2.6}$$

The generator and the discriminator are trained together, with k optimisation steps of D following one optimisation step of G. In other words, G and D contest with each other in a two-player zero-sum game with objective function $V(\theta_G, \theta_D)$:

$$\min_{\theta_G} \max_{\theta_D} V(\theta_G, \theta_D) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{2.7}$$

In [GPAM+14] it has been shown that given an optimal discriminator D^*, minimising Eq. (2.7) is equivalent to minimising:

$$C(\theta_G) = KL\left(p_{data}\middle\|\frac{p_{data}+p_g}{2}\right) + KL\left(p_g\middle\|\frac{p_{data}+p_g}{2}\right) - \log(4). \tag{2.8}$$

Here, KL denotes the Kullback–Leibler divergence. This is essentially the Jensen-Shannon (JSD) divergence between the data and model distribution:

$$C(\theta_G) = 2 \cdot JSD(p_{data}||p_g) - \log(4). \tag{2.9}$$

Given that the Jensen–Shannon divergence between two distributions is always non-negative and zero only when the distributions are equal, the optimal generator is given when $p_g = p_{data}$, as the generative model perfectly replicates the training data generating process.

Since the seminal work, numerous alternatives to traditional GAN training have been proposed. These works focus mainly on the objective function and the measure of divergence between the model distribution and actual distribution, as they aim to improve the stability of learning and mitigate common problems of GANs, such as mode collapse. Here, we present a short overview of some well-known variations, namely Wasserstein GAN (WGAN) [ACB17] and Wasserstein GAN with gradient penalty (WGAN GP) [GAA+17], Least Squares GAN (LSGAN) [MLX+17] and Geometric GAN (GAN with Hinge loss) [LY17].

WGAN [ACB17] WGAN does not use the Jensen-Shannon (JSD) to measure divergence, instead it uses the Earth-Mover (EM) distance that is also known as Wasserstein distance. Earth-Mover distance, or Wasserstein-1 between distributions p_{data} and p_g is defined as:

$$W(p_{data}, p_g) = \inf_{\gamma \in \Pi(p_{data}, p_g)} \mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})\sim\gamma}[||\mathbf{x} - \tilde{\mathbf{x}}||_1]. \tag{2.10}$$

$\Pi(p_{data}, p_g)$ denotes the set of all joint distributions $\gamma(\mathbf{x}, \tilde{\mathbf{x}})$ with marginal distributions p_{data} and p_g. Intuitively, $\gamma(\mathbf{x}, \tilde{\mathbf{x}})$ reflects how much "mass" has to be transferred from $\tilde{\mathbf{x}}$ to $\mathbf{x}$ for converting the distribution p_g into p_{data}. The Earth-Mover distance corresponds to the "cost" of the best transport plan. Although the infimum in Eq. (2.11) is highly intractable, the Kantorovich-Rubinstein duality gives us:

$$W(p_{data}, p_g) = \sup_{||f||_L \leq 1} \mathbb{E}_{\mathbf{x}\sim p_{data}}[f(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}}\sim p_g}[f(\tilde{\mathbf{x}})], \tag{2.11}$$

that is the supremum over 1-Lipschitz continuous functions $f : \mathcal{X} \to \mathbb{R}$. Therefore, if we parameterise the family of functions f with a discriminator network D, we could consider solving the optimisation problem:

$$\max_{\theta_D} \mathbb{E}_{\mathbf{x}\sim p_{data}}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{x})}[D(G(\mathbf{z}))], \tag{2.12}$$

while making sure that D is 1-Lipschitz, or in general K-Lipschitz for some $K \in \mathbb{R}$. To enforce this constraint, WGAN clamps the discriminator's parameters θ_D to a fixed box in order to restrict the weights of the discriminator within a certain range $[-c, c]$. At the same time, the generator G acts as an adversary and tries to minimise the same value, which gives the following objective:

$$\min_{\theta_G} -\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z}))]. \tag{2.13}$$

WGAN-GP [GAA+17] The performance of WGAN is very sensitive to the hyperparameter c that controls the discriminator's parameters range and ensures the K-Lipschitz continuity for D. Weight clipping decreases the learning capacity of D and limits its ability to model complex functions. Instead of weight clipping, WGAN-GP relies on gradient penalty in order to impose the Lipschitz continuity. Now, objective of network D becomes:

$$\min_{\theta_D} \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z}))] - \mathbb{E}_{\mathbf{x}\sim p_{data}}[D(\mathbf{x})] + \lambda_{GP}\mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{\mathbf{x}}}}[(||\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}})||_2 - 1)^2], \tag{2.14}$$

where $\hat{\mathbf{x}} = a\mathbf{x} + (1-a)\tilde{\mathbf{x}}$ is computed as a random linear combination of the real and generated samples $\mathbf{x}$ and $\tilde{\mathbf{x}} = G(\mathbf{z}), \mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$, with $a \sim \mathcal{U}(0,1)$. The hyperparameter λ_{GP} controls the contribution of the gradient penalty into the total cost function.

LSGAN [MLX+17] The objective functions for LSGAN are defined as follows:

$$\min_{\theta_D} \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{data}}[(D(\mathbf{x}) - b)^2] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[(D(G(\mathbf{z})) - a)^2] \tag{2.15}$$

$$\min_{\theta_G} \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[(D(G(\mathbf{z})) - c)^2]. \tag{2.16}$$

Here, a and b are the labels for synthetic and real data respectively, while c denotes the label that G expects D to assign to generated data. It has been shown [MLX+17] that if $b - c = 1$ and $b - a = 2$, then the minimisation of the objective functions above is the same as minimizing the Pearson χ^2 divergence between $p_{data} + p_g$ and $2p_g$.

Geometric GAN [LY17] The GAN hinge loss is defined as

$$\min_{\theta_D} -\mathbb{E}_{\mathbf{x}\sim p_{data}}[\min(0, -1 + D(\mathbf{x}))] - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\min(0, -1 - D(G(\mathbf{z})))] \tag{2.17}$$

$$\min_{\theta_G} \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[D(G(\mathbf{z}))]. \tag{2.18}$$

2.2.2 GAN-based Image and Video Translation

Much research around GANs has been focused on image generation. Various methods have been introduced for generating random photo-realistic image data from latent codes, indistinguishable from samples in the training set [KALL17, KLA19]. However, these models do not offer a straightforward way to control the generative process, as the properties of the latent space is poorly understood yet, and generators operate as black boxes. Controlling the output of generators is of paramount importance when it comes to exploiting the full potential of GANs. Early works on conditional image synthesis relied on auto-encoders that learned to synthesise target images from conditional inputs with the assistance of L1 or L2 pixel distance for reconstruction [DTSB15]. Given that these models show a tendency to produce blurry results, later attempts [GEB16, DB16, JAFF16] capitalised on perceptual similarity distances to calculate the error between synthesised and real data, in a higher level feature representation that is extracted by a pre-trained neural network. Nonetheless, this approach does not guarantee that generated images look natural and photo-realistic.

Conditional GAN Rather than minimising the distance between the synthetic and ground images, conditional GANs [MO14] try to bring the conditional distribution of outputs as close as possible to the training data distribution, for given inputs. Although their objective is not reconstruction per se, cGANs constitute a way to explicitly control synthesis of realistic data samples. GANs can be viewed as a conditional models, by providing the generator network with an extra input $\mathbf{y}$ that guides synthesis. The conditional input $\mathbf{y}$ may be a class label, an embedding vector, an image, or any form of data related to the problem. cGANs learn a mapping $G : \{\mathbf{z}, \mathbf{y}\} \rightarrow \tilde{\mathbf{x}}$, as the latent code $\mathbf{z}$ is combined with input $\mathbf{y}$ into a joint representation. The objective of cGAN is:

$$\min_{\theta_G} \max_{\theta_D} V(\theta_G, \theta_D) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}|\mathbf{y}))] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z}|\mathbf{y})))]. \tag{2.19}$$

Training of cGANs can be stabilised by combining the adversarial objective with a per-pixel L1 loss or a perceptual distance loss. In the following, we present an overview of methods combining cGANs with reconstruction losses for the task of image-to-image or video-to-video translation.

pix2pix [IZZE17] This is arguably one of the pioneering works on image-to-image translation with the assistance of cGANs [MO14]. Here, we assume that have access to paired training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, where $\mathbf{x}_i$ are images of the input domain and $\mathbf{y}_i$ are the corresponding image pairs of the target domain. Images of the input domain are usually abstract representations of the target images, such as segmentation label maps, sketches or edge maps, nonetheless, in principle the method generalises to any input and output image domain. Here, the goal is to train a generator G that translates conditional inputs $\mathbf{x}_i$ to outputs $\mathbf{y}_i$, which cannot be distinguished from "real" images by an adversarially trained discriminator D. The objective of network G can be expressed as:

$$\min_{\theta_G} \mathcal{L}_{adv}^{G} + \lambda_{L1} \mathcal{L}_{L1}^{G}, \tag{2.20}$$

where $\mathcal{L}_{adv}^{G}$ is the adversarial or GAN objective:

$$\mathcal{L}_{adv}^{G} = \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}), (\mathbf{x}, \mathbf{y}) \sim p_{data}(\mathbf{x}, \mathbf{y})}[\log(1 - D(\mathbf{x}, G(\mathbf{z}, \mathbf{x})))] \tag{2.21}$$

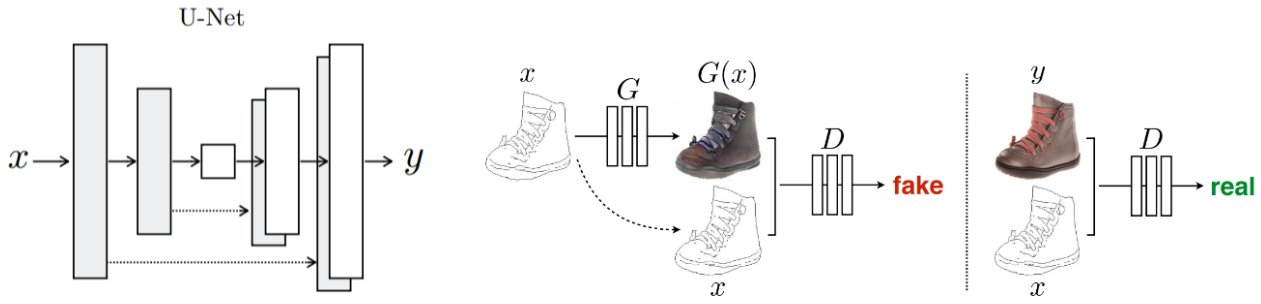
and $\mathcal{L}_{L1}^{G}$ is the reconstruction pixel loss term:

$$\mathcal{L}_{L1}^{G} = \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}), (\mathbf{x}, \mathbf{y}) \sim p_{data}(\mathbf{x}, \mathbf{y})}[||\mathbf{y} - G(\mathbf{z}, \mathbf{x})||_1]. \tag{2.22}$$

The hyper-parameter λ_{L1} regulates the contribution of the L1 pixel loss to the overall objective function for G. L1-distance is used instead of L2 as it results in less blurring. Following the setup of cGAN, the discriminator learns to detect samples produced by the generator, by observing a generated or real sample and the corresponding conditional input $\mathbf{x}$. This gives the training objective for network D:

$$\max_{\theta_D} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim p_{data}(\mathbf{x}, \mathbf{y})}[\log(D(\mathbf{x}, \mathbf{y}))] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z}), (\mathbf{x}, \mathbf{y}) \sim p_{data}(\mathbf{x}, \mathbf{y})}[\log(1 - D(\mathbf{x}, G(\mathbf{z}, \mathbf{x})))]. \tag{2.23}$$

Without the latent codes $\mathbf{z}$, the generator would still learn a mapping from $\mathbf{x}$ to $\mathbf{y}$ in a deterministic manner. The authors of [IZZE17] reported that G actually learns to ignore the noise $\mathbf{z}$, therefore in practice they provided noise only in the form of dropout, applied on several layers of the generator network, yet this resulted only in minor stochasticity. For network G, they use an encoder-decoder architecture, as the input is passed through a series of layers that progressively



(a) Net with skip connections (b) An overview of pix2pix data flow during training

Figure 2.4: (a) pix2pix generator for image-to-image translation follows an encoder-decoder architecture with skip connections. (b) The discriminator learns to classify between synthesised images $G(x)$ and real samples y, by observing also the conditional input x. Image adapted from pix2pix [IZZE17].

down-sample it, up to a bottleneck layer, where the process is reversed to up-sampling. They add skip connections, to enable the generator circumvent intermediate layers, following the general shape of a "U-Net", as seen in Fig. 2.4. Given that L1 is sufficient to synthesise course low-frequency information the adversarial loss is mainly used to learn high-frequency visual details, the authors restrict the discriminator's attention to local image patches. To that end, they propose PatchGAN, a discriminator that inspects samples at the scale of patches. Thus, network D outputs a different scalar number for each small image patch, which determines whether the patch looks real or not.

pix2pixHD [WLZ+18b] In a follow up work, [WLZ+18b] improves the pix2pix framework by using a coarse-to-fine generator, a multi-scale discriminator, as well as a perceptual loss and a feature matching loss based on the discriminator. Their novel architecture translates semantic label maps into realistic images. First, they train a generative network G_1 on lower resolution images. Then, another network G_2 is appended to G_1 (to the front and back end of G_1) and the two networks that comprise G are trained jointly on higher resolution images. An overview of G is presented in Fig. 2.5. Furthermore, they propose to use three discriminators that have an identical network structure but operate at different image scales. With LSGAN [MLX+17] setup and three discriminators the adversarial loss for network G becomes:

$$\mathcal{L}_{adv}^{G} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{data}(\mathbf{x},\mathbf{y})}[\sum_{k=1}^{3}(D_k(\mathbf{x}, G(\mathbf{x})) - 1)^2]. \tag{2.24}$$

Please note that in contrast to pix2pix [IZZE17], the generator does not receive a latent code as input. Moreover, instead of an L1 pixel loss, they propose a feature matching loss using the discriminators. For that, they extract features from multiple layers of each discriminator and formulate a loss term by matching these intermediate representations from the real and synthesised images:

$$\mathcal{L}_{fm}^{G} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{data}(\mathbf{x},\mathbf{y})}[\sum_{k=1}^{3}\sum_{j=1}^{N_l}\frac{1}{N_j}||D_k^{(j)}(\mathbf{x},\mathbf{y}) - D_k^{(j)}(\mathbf{x}, G(\mathbf{x}))||_1]. \tag{2.25}$$

Here, N_l is the number of layers in the discriminators and $N_j = H_j \times W_j$ is the number of elements for each layer j. Even though the feature matching loss acts as a perceptual loss,

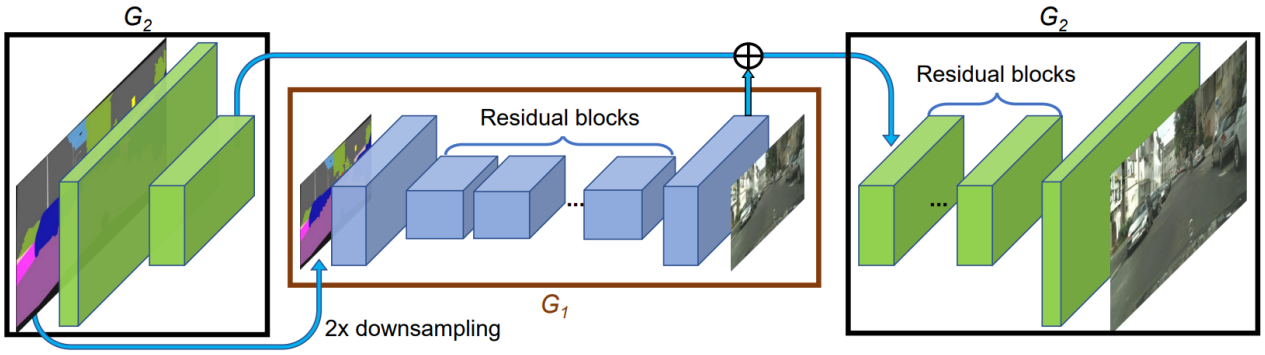


Figure 2.5: The architecture of pix2pixHD generator G, consists of G_1 and G_2. First, G_1 is trained on lower resolution images and then G_2 is appended, with the entire system trained on images of higher resolution. The input to the residual blocks of G_2 is the sum of the feature map from G_2 and the last feature map from G_1. Image adapted from [WLZ+18b].

it has been shown that adding an extra perceptual loss based on a pre-trained VGG network [SZ14] slightly improves the results:

$$\mathcal{L}_{vgg}^{G} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{data}(\mathbf{x},\mathbf{y})}[\sum_{j=1}^{M_l} \frac{1}{M_j} ||VGG^{(j)}(\mathbf{y}) - VGG^{(j)}(G(\mathbf{x}))||_1]. \tag{2.26}$$

M_l denotes the number of layers in the VGG network and M_j denotes the number of elements in each layer. The final objective for G is formulated as:

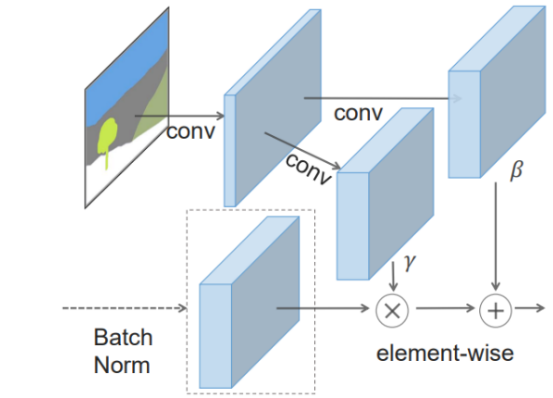
$$\min_{\theta_G} \mathcal{L}_{adv}^{G} + \lambda_{fm}\mathcal{L}_{fm}^{G} + \lambda_{vgg}\mathcal{L}_{vgg}^{G}, \tag{2.27}$$

while each of the three discriminators D_k is trained under the following objective:

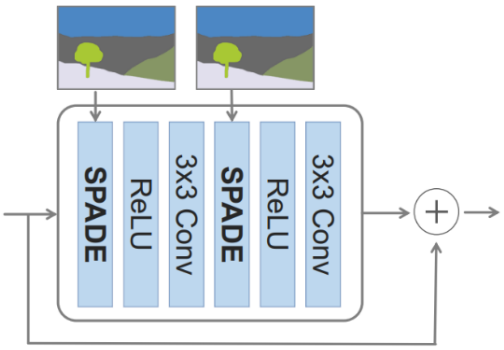
$$\max_{\theta_{D_k}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{data}(\mathbf{x},\mathbf{y})}[(D_k(\mathbf{x},\mathbf{y}) - 1)^2] + \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{data}(\mathbf{x},\mathbf{y})}[D_k(\mathbf{x}, G(\mathbf{x}))^2]. \tag{2.28}$$

Additionally, the authors propose instance boundary maps as an extra input to G, as they provide important information of the objects boundaries, not available in the semantic label maps. Lastly, in order to generate diverse images and allow instance-level control, they propose adding a low-dimensional feature map as an additional input to the generator network. These feature maps are extracted from real images with another encoder-decoder network, learned jointly with the GAN.

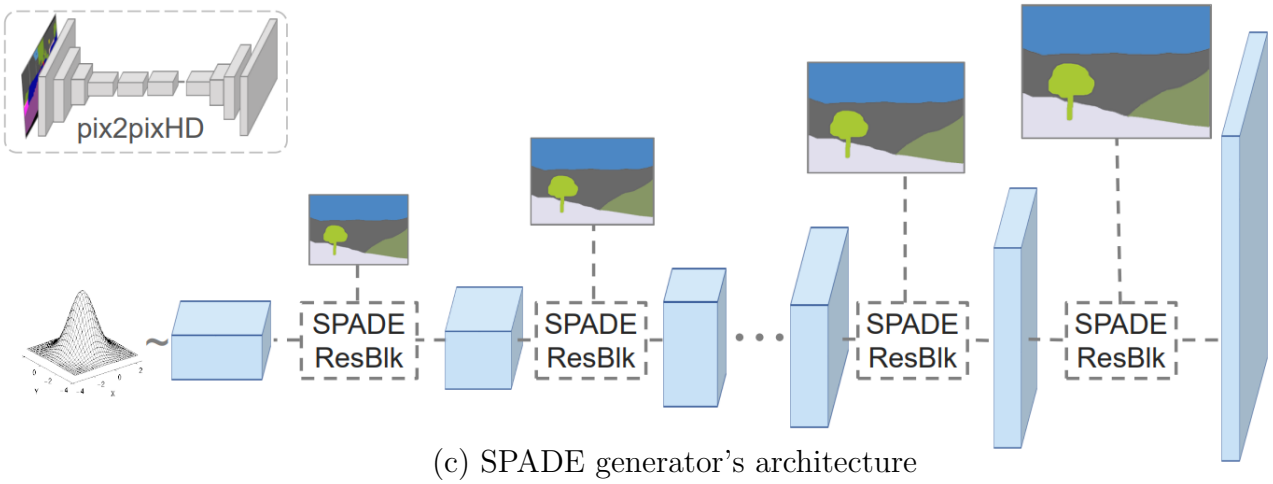
SPADE [PLWZ19] proposes a spatially-adaptive normalisation layer for synthesising photo-realistic images given a semantic segmentation map as input. As opposed to previous methods [IZZE17, WLZ+18b] that directly "feed" the segmentation map as input to the generator network, this work suggests that it is more effective to use the input image in order to modulate the activations in the normalization layers, through a spatially-adaptive learned transformation. This design choice removes the need for an encoding part in G, which is just a decoder. This architecture of G can receive a random latent vector as input, providing a straightforward way to multi-modal synthesis. However, in practise the authors propose an encoder that captures the style of a given image into a random vector. Assuming a training batch of size N, let $\mathbf{h}^{(j)} \in \mathbb{R}^{N\times C^{(j)}\times H^{(j)}\times W^{(j)}}$ denote the activation of the j-th layer of the decoder with $C^{(j)}$ channels and spatial size $H^{(j)} \times W^{(j)}$. As with batch normalization [IS15], this activation is normalized



(a) Spatially-adaptive normalisation (SPADE)



(b) SPADE ResNet block (ResBlk)



(c) SPADE generator's architecture

Figure 2.6: Different from pix2pixHD [WLZ+18b], the generator of [PLWZ19] (a) is a decoder that consists of SPADE ResNet blocks (b) and up-sampling layers. Each ResNet block contains two SPADE layers (c), which normalise their input with modulation parameters $\gamma(\mathbf{x})$ and $\beta(\mathbf{x})$ that are computed with convolutional neural networks from semantic segmentation maps (generator's conditional input $\mathbf{x}$). The image is adapted from [PLWZ19].

in a channel-wise manner and then modulated with a learned scale and bias. Given a segmentation map input $\mathbf{x}$, the normalised activation value at $n \in N, c \in C^{(j)}, h \in H^{(j)}, w \in W^{(j)}$ is given by:

$$\mathbf{h}'^{(j)}_{n,c,h,w} = \gamma^{(j)}_{n,c,h,w}(\mathbf{x}) \frac{\mathbf{h}^{(j)}_{n,c,h,w} - \mu^{(j)}_c}{\sigma^{(j)}_c} + \beta^{(j)}_{n,c,h,w}(\mathbf{x}) \tag{2.29}$$

where $\mu^{(j)}_c$ and $\sigma^{(j)}_c$ are the mean and standard deviation of the activation for channel c:

$$\mu^{(j)}_c = \frac{1}{NH^{(j)}W^{(j)}} \sum_{n,h,w} \mathbf{h}^{(j)}_{n,c,h,w}, \tag{2.30}$$

$$\sigma^{(j)}_c = \sqrt{\frac{1}{NH^{(j)}W^{(j)}} \sum_{n,h,w} \left(\left(\mathbf{h}^{(j)}_{n,c,h,w}\right)^2 - \left(\mu^{(j)}_c\right)^2 \right)}. \tag{2.31}$$

The functions $\gamma^{(j)}(\mathbf{x})$ and $\beta^{(j)}(\mathbf{x})$ are parameterised with a convolutional neural network, that receives as input the segmentation map $\mathbf{x}$ and learns to produce the scale and bias tensors.

The spatially-adaptive normalisation layer described above, namely SPADE, is illustrated in Fig. 2.6(a). SPADE layers are used to form SPADE ResNet blocks, which constitute the fundamental building blocks of the generator, as shown in Fig. 2.6(c). The generator is trained alongside the same multi-scale discriminator and loss function employed in pix2pixHD, except that LSGAN loss has been replaced with a Hinge loss from Geometric GAN [LY17].

vid2vid Unlike image-to-image translation with paired data [IZZE17, WLZ+18b, PLWZ19], the counterpart problem of video-to-video translation has received less attention. vid2vid [WLZ+18a] is one of the few GAN-based attempts to solve this problem, by learning a function that can convert an input video to an output video of a different domain. Directly applying existing image translation algorithms like pix2pix to a sequence of frames often results in temporally incoherent output videos, as the generator produces frames independent from each other. In contrast, the authors of [WLZ+18a] suggest that a sequential generator that learns to predict optical flow alongside a temporal discriminator performs better in terms of temporal stability and human perception of video realism. Let $\mathbf{x}_{1:T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ be a sequence of T input frames and $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$ be the corresponding sequence of output frames, which form the training data. The conditional input video $\mathbf{x}_{1:T}$ may be a sequence of edge maps, semantic segmentation maps, etc., while $\mathbf{y}_{1:T}$ is typically a real video. vid2vid performs frame synthesis with the assistance of a conditional sequential generator, as frame t is produced by the rule:

$$\tilde{\mathbf{y}}_t = G(\mathbf{x}_{t-L:t}, \tilde{\mathbf{y}}_{t-L:t-1}). \tag{2.32}$$

In other words, the generation of $\tilde{\mathbf{y}}_t$ is conditioned on the current input frame $\mathbf{x}_t$, the past L input frames $\mathbf{x}_{t-L:t-1}$, as well as the previously L generated frames $\tilde{\mathbf{y}}_{t-L:t-1}$. The generator is applied recursively to create a video sequence. Since consecutive frames share a significant amount of information, they design a generator that leverages on that by predicting a dense optical flow field for warping the previous synthetic frame. That is, G outputs a intermediate prediction for frame t, namely $\hat{\mathbf{y}}_t$, as well as an optical flow $\tilde{\mathbf{w}}_{t-1}$ estimation and an occlusion mask $\tilde{\mathbf{m}}_t$. Then, the generated frame t is computed as:

$$\tilde{\mathbf{y}}_t = G(\mathbf{x}_{t-L:t}, \tilde{\mathbf{y}}_{t-L:t-1}) = \tilde{\mathbf{m}}_t \odot \hat{\mathbf{y}}_t + (1 - \tilde{\mathbf{m}}_t) \odot \tilde{\mathbf{w}}_{t-1}(\tilde{\mathbf{y}}_{t-1}), \tag{2.33}$$

where $\tilde{\mathbf{w}}_{t-1}$ is the optical flow estimated by network G, from frame $\tilde{\mathbf{y}}_{t-1}$ to $\tilde{\mathbf{y}}_t$, while $\tilde{\mathbf{w}}_{t-1}(\tilde{\mathbf{y}}_{t-1})$ denotes the operation of warping the previously generated frame $\tilde{\mathbf{y}}_{t-1}$, according to the predicted optical flow. In addition, $\tilde{\mathbf{m}}_t \in [0, 1]^{H \times W}$ is the occlusion mask that determines how to blend the newly generated pixels $\hat{\mathbf{y}}_t$ with the warped image. An overview of vid2vid generator is depicted in Fig. 2.7. As can be seen, the authors adopt a coarse-to-fine generator approach, similar to the method of [WLZ+18a]. The network G is trained adversarially with two separate discriminators: one image discriminator D_I and one temporal discriminator D_V. The image discriminator D_I acts as a typical conditional discriminator that receives either a real or a synthetic frame along with the corresponding conditional input image and learns to distinguish between $(\mathbf{x}_t, \mathbf{y}_t)$ and $(\mathbf{x}_t, \tilde{\mathbf{y}}_t)$. A multi-scale [WLZ+18a] PatchGAN [IZZE17] architecture is used for D_I. In addition, the temporal discriminator ensures that consecutive output frames resemble the temporal dynamics of a real video, given the same optical flow. Let $\mathbf{w}_{t-K:t-1}$ be the K ground truth optical flows for $K + 1$ frames $\mathbf{y}_{t-K:t}$. The discriminator D_V is trained to tell apart the real frames $\mathbf{y}_{t-K:t}$ from the generated ones $\tilde{\mathbf{y}}_{t-K:t}$, while conditioning on the true optical flow, therefore, distinguish $(\mathbf{w}_{t-K:t-1}, \mathbf{y}_{t-K:t})$ from $(\mathbf{w}_{t-K:t-1}, \tilde{\mathbf{y}}_{t-K:t})$. In practice, D_V looks at different frame rates of the video data, to promote both short-term and long-term consistency. Aside from the adversarial losses, G is optimised using a perceptual loss and a

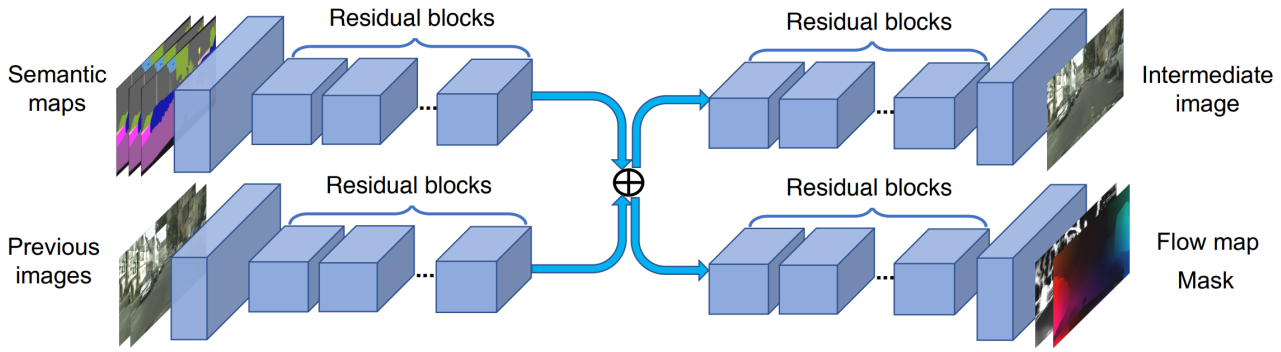


Figure 2.7: The architecture of vid2vid generator. The conditional input $\mathbf{x}_{t-L:t-1}$ (e.g. semantic maps) is processed by a sequence of down-sampling and residual blocks. A similar encoding branch processes the past synthetic frames $\tilde{\mathbf{y}}_{t-L:t-1}$ (previous images). The two resulting feature maps are added together and passed through two identical decoding branches, one that hallucinates $\hat{\mathbf{y}}_t$ (intermediate image) and another that predicts the optical flow $\tilde{\mathbf{w}}_{t-1}$ (flow map) for warping $\tilde{\mathbf{y}}_{t-1}$ and the occlusion mask $\tilde{\mathbf{m}}_t$ (mask). Image adapted from vid2vid [WLZ+18a].

discriminator-based feature matching loss that ensure training stability and enforce ground truth reconstruction, as in pix2pixHD [WLZ+18b]. Finally, an additional loss term related to optical flow helps G to learn the task of flow estimation:

$$\mathcal{L}^G_{flow} = \mathbb{E}_{t\sim U\{2,T\}}[||\mathbf{w}_{t-1} - \tilde{\mathbf{w}}_{t-1}||_1 + ||\tilde{\mathbf{w}}_{t-1}(\mathbf{y}_{t-1}) - \mathbf{y}_t||_1]. \tag{2.34}$$

2.3 Video Portrait Synthesis and Reenactment

2.3.1 Face Reenactment and Visual Dubbing

The majority of early attempts to controllable face synthesis focus on the task of *face reenactment* [GVR+14, TZN+15, TZS+16], which aims to transfer the facial expressions performed in a driving video to the face of a target person. One of the first and most impressive face reenactment systems is Face2Face [TZS+16], which manipulates a target video footage by rewriting only the facial region of the given frame sequence, while keeping the remaining parts (e.g. hair, body, background) unchanged. *Visual dubbing* is a specific instance of face reenactment with the objective to modify only the mouth movements of a target video in order to match the speech performance of a source actor. Such methods are either performance-driven [GVS+15] or audio-driven [SSKS17], which produce very accurate lip motions but cannot control facial expressions. The aforementioned approaches are not able to control the head pose of the subject and in many cases struggle to synthesize photo-realistic mouth movements. On the contrary, a more recent graphics-based approach, namely HeadOn [TZT+18], performs full head reenactment that includes pose and gaze transfer based on RGB-D video data.

2.3.2 Full Head Reenactment

Deep Video Portraits (DVP) [KGT+18] is allegedly one of the earliest learning-based full head reenactment methods, as it utilises an image translation network that manages to fully transfer the head motions of the driver to the source, including eye gaze. This method relies on parametric 3D face representations with strong statistical priors, such as 3DMMs [BV99]. Despite its remarkable results, DVP is an image-based *person-specific* model that is optimised on long videos of the target identity and is not capable to adapt and generalise to new identities without re-training.

On the other hand, *person-generic* methods are able to adapt their generative process on the appearance of new source identities, even if they are not introduced to them during training. The technique presented in [AECOKC17] is among the first person-agnostic attempts, which is able to slightly animate the entire head with the application of 2D warping on a reference image. In the following years, numerous identity-generic talking head synthesis systems that condition image synthesis on 2D facial landmarks have been proposed [ZSBL19, ZISL20, WLT+19, HKK+20]. The few-shot GAN-based framework introduced in [ZSBL19] consists of an identity embedding network and an image generator with adaptive instance normalization (AdaIN) layers [HB17] that receive the embeddings and modulate synthesis on the appearance of the target individual. The authors present a training strategy consisting of a meta-learning stage which involves optimisation on a multiperson database, followed by fine-tuning using an few images of an unseen face. In a similar fashion, the video-based few-shot vid2vid [WLT+19] method employs a sequential generator equipped with dynamic SPADE layers [PLWZ19], enabling to adapt synthesis on the appearance of new reference images. Following up, a SPADE-based system is presented in [ZISL20], which achieves real-time one-shot talking head synthesis on mobile devices. Their generator network focuses on the foreground and disregards background information. Instead of representing faces with 2D landmarks, 3D-based approaches such as Warp-Guided GANs [GSZ+18] and MarioNETte [HKK+20] depend on 3D landmarks to model facial performance. As opposed to the prior works, MarioNETte [HKK+20] applies disentanglement on identity and expression based on PCA of 3D facial landmarks. In this way, it alleviates the identity mismatch problem during cross-identity reenactment.

There is a considerable amount of research that involves person-generic *model-free* synthesis of human heads [WKZ18, SLT+19a, SLT+19b, WML21]. X2Face [WKZ18] is one of the pioneering approaches that does not rely on any landmarks or 3D face priors. It is able to animate human faces from multiple modalities, such as images, audio, and pose codes. In an attempt to address motion transfer for arbitrary objects, Monkey-Net [SLT+19a] proposes a framework that is made up of a 2D key-point detector that is jointly trained with a motion prediction and motion transfer network. 2D key-points are learned from data in an unsupervised way and placed to meaningful regions of the faces in order to enable reliable optical flow approximation for motion transfer. In the follow-up work, [SLT+19b] proposes a first order approximation of optical flow, based again on learnable key-points, which yields very promising motion transfer results. However, during cross-identity motion transfer this method uses relative key-points to preserve the identity in the reference image, which relies on the assumption that the face in the first driving frame has the same pose with the reference face. Another more recent work, namely face-vid2vid [WML21], applies first order motion approximation on 3D facial key-points. The 3D points enables this model to perform free-view reenactment, as it can control pose by rotating the 3D key-points. The entire system consists of two other networks, one that predicts canonical key-points and one that estimates pose and expression deformations. This

Table 2.1: Checklist of key features and design choices of state-of-the-art head synthesis systems that support full head animation and reenactment

Method	Person Generic	N-shot	Face Modeling	Pose Control	Preserve Identity	Gaze Control	Warping Flow
DVP [KGT+18]	✗	-	3DMM	✓	✓	✓	✗
Head2Head [KDRZ20]	✗	-	3DMM	✓	✓	✓	✗
Zakharov et al. [ZSBL19]	✓	✓	2D landmarks	✗	✗	✗	✗
few-shot vid2vid [WLT+19]	✓	✓	2D landmarks	✗	✗	✗	✗
Zakharov et al. [ZISL20]	✓	✗	2D landmarks	✗	✗	✗	✗
MarioNETte [HKK+20]	✓	✓	3D landmarks	✗	✓	✗	learned
Warp-Guided GAN [GSZ+18]	✓	✗	3D landmarks	✓	✗	✗	crafted
X2Face [WKZ18]	✓	✓	embeddings	✗	✗	implicit	learned
Monkey-Net [SLT+19a]	✓	✗	2D key-points	✗	✗	implicit	approx.
Siarohin et al. [SLT+19b]	✓	✗	2D key-points	✗	✗	implicit	approx.
face-vid2vid [WML21]	✓	✗	3D key-points	✓	✓	implicit	approx.
HeadGAN [DZS21]	✓	✗	3DMM	✓	✓	✗	learned
Free-HeadGAN	✓	✓	3D landmarks	✓	✓	✓	learned

formulation helps to tackle the identity preservation problem. In Table 2.1 we present a detailed summary of the key features and design choices of state-of-the-art systems for reenactment and motion transfer, as discussed in this section. For a more complete comparison, we included the methods presented in this thesis, and more specifically Head2Head [KDRZ20] (Chapter 3), HeadGAN [DZS21] (Chapter 5) and Free-HeadGAN (Chapter 6).

2.3.3 Neural Representations for Scenes and Faces

With the introduction of Neural Radiance Fields (NeRFs) [MST+20], much research has been focused on neural scene representations [MBRS+21, PCPMMN21], with various attempts to model human faces. More specifically, Nerfies [PSB+21] and HyperNeRF [PSH+21] have shown incredible results for reconstructing non-rigid scenes of moving heads, as they are built upon dynamic variations of NeRF [PCPMMN21, TTG+21]. Despite their impressive generative ability, such systems are not able to control neither head poses nor facial movements. In a different direction, NerFACE [GTZN21] proposes to control a dynamic NeRF with the assistance of expression blendshapes. However, NerFACE handles tracked head pose parameters as camera viewpoints, which leads to severe artifacts when rendering upper body pixels. In a similar line of work, AD-NeRF [GCL+21] proposes an audio-driven model, based on acoustic features extracted with DeepSpeech [HCC+14, AAA+16]. The authors resort in body part segmentation priors and two separate NeRFs to mitigate torso-related problems. Despite their promising results, NerFACE [GTZN21] and AD-NeRF [GCL+21] are particularly slow in terms of frame rendering speeds.

Part I

Person-Specific Reenactment with Complete Head Control

Chapter 3

Head2Head: Video-based Full Head Reenactment

A growing line of research revolves around *face reenactment* and *full head reenactment*. Initially, these problems were solely studied by the computer graphics community, which focused mainly on face reenactment [GVR+14, TZN+15, TZS+16]. Face reenactment methods design solutions for transferring the expressions from a source actor to a target video, by manipulating the target video footage within the inner facial area of the subject, without modifying the person's head position or eye gaze. More recent algorithms aim to transfer the complete head movements of the source to the target identity, including eye gaze, a problem know as full head reenactment or *controllable video portrait synthesis* [TZT+18, KGT+18]. A video portrait is defined as a sequence of frames, depicting a person that performs diverse head movements and facial expressions. The person's entire head is contained within the frame's borders, along with a small part of the upper body, i.e. neck and torso, while the subject usually stands in front of a static background. Controllable video portrait synthesis is more challenging than face reenactment, since it involves generating the entire frames, rather than a particular region of interest. Conditional GANs for image-to-image translation have shown very promising results towards this direction, as they are able to learn a mapping from classical computer graphics renderings to photo-realistic images [KGT+18]. For this task, typically a separate neural network is optimised for each target actor, using a long training video. Nonetheless, given that image-to-image translation models synthesise frames independently from each other, in some cases the generated videos exhibit *temporal incoherence* between frames. Other approaches have been proposed for controlling entire heads with the assistance of few-shot learning, where a small number of reference frames are accessible [ZSBL19, WLT+19]. Many of these methods guide synthesis with 2D sketches drawn from facial landmarks [ZSBL19, WLT+19, ZISL20], which usually leads to train-test gaps and *identity preservation* issues in cross-identity reenactment, as landmarks contain appearance-related information of the source actor. The problem is even more prominent in cases where the source facial shape is significantly different from the facial shape of the target identity.

While a learning-based solution to full head reenactment has been already studied in Deep Video Portraits [KGT+18], here we approach the problem from a video-based perspective. Furthermore, we show how our person-specific method can be extended and adapted to a generic model for multiple identities, using concepts from few-shot learning [ZSBL19]. In this work, we extend prior reenactment systems in a number of dimensions:

1. We propose **Head2Head** [KDRZ20], a method for full head reenactment that is based on video neural rendering. Our system is able to reenact the performance of any source actor, by generating video footage of a specific target identity. We condition frame synthesis on a short sequence of images with semantic facial information, and more specifically PNCCs [ZLL+16], rather than face renderings that include texture information [KGT+18]. PNCC images are acquired using a fast video-based 3D face reconstruction algorithm.

2. We argue that full head reenactment is inherently a video-to-video translation task, and thus applying image-to-image translation models without modeling temporal dynamics might result in temporally incoherent sequences of frames. To that end, we adopt a conditional GAN with a sequential generator, as well as a temporal discriminator. Our analysis, which includes a user study, an automated study and an ablation study, reveals the significance of video-based neural rendering.

   In addition, we give special attention to the mouth area, which is very crucial when it comes to the photo-realism of the reenacted face. For that, we propose a discriminator network dedicated to the mouth area of the target subject, which results in finer details within the mouth cavity and teeth.

3. Lastly, we extend our person-specific Head2Head framework to generate multiple identities with the assistance of an identity encoding network, a generator with adaptive instance normalisation (AdaIN) layers [HB17] and a few-shot learning strategy. Our extended model is first trained on a large video dataset with multiple identities and then fine-tuned with the video footage of an unseen person, using a few hundreds frames.

3.1 Face Tracking

Typically, a reenactment system consists of two distinct components: a module that tracks the movements of the source actor, and a system that utilises the tracked information to generate the footage of the target identity. Considering that full head reenactment aims to transfer all head movements from the source to the target actor, one needs a tracking module capable of recovering *facial expressions*, *head pose*, as well as the *eye gaze*. Sparse facial landmarks encode both expression and pose information, and thus have been widely used to model facial behaviour in numerous methods for reenactment and motion transfer [ZSBL19, ZISL20, SLT+19a, SLT+19b]. Nonetheless, they also carry identity related information, which during cross-identity reenactment gives rise to the identity preservation problem and affects the quality of generated samples crucially. We solve this problem by utilising a 3D face reconstruction algorithm instead, which relies on the properties of 3DMMs, such as expression and identity (shape) disentanglement, and 3DMM fitting [BV99, PKA+09, BRP+18, CWZ+13, LBB+17]. In this section, we describe the face tracking pipeline we adopted in Head2Head [KDRZ20], which recovers expression, pose and gaze signals from a given sequence of frames.

3.1.1 3D Facial Shape and 3DMMs

We seek to obtain an accurate prediction of the 3D facial shape and at the same time separate the identity and expression related information of a portrayed subject in a given video. Towards

this aim, we benefit from the prior knowledge in the field of face modeling and make use of 3D Morphable Models for reconstructing the face of the subject that appears in a video. Given an input video with T frames $\mathbf{y}_{1:T} = \{\mathbf{y}_t | t = 1, \ldots, T\}$, the face recovering algorithm returns the camera parameters $\mathbf{p}^c_{1:T} = \{\mathbf{p}^c_t \in \mathbb{R}^7 | t = 1, \ldots, T\}$ that account for scale, rotation and translation, as well as the shape parameters, which are further divided into identity shape parameters $\mathbf{p}^{id}_{1:T} = \{\mathbf{p}^{id}_t \in \mathbb{R}^{n_{id}} | t = 1, \ldots, T\}$ and expression shape parameters $\mathbf{p}^{exp}_{1:T} = \{\mathbf{p}^{exp}_t \in \mathbb{R}^{n_{exp}} | t = 1, \ldots, T\}$.

Using a 3DMM for identity shape [BRP+18] and another for expression [CWZ+13], the face depicted at frame t can be represented by a 3D mesh with vertices $\mathbf{s}_t = [x_1, y_1, z_1, ..., x_N, y_N, z_N]^T \in \mathbb{R}^{3N}$, using the following formula:

$$\mathbf{s}_t = \mathcal{S}(\mathbf{p}^{id}_t, \mathbf{p}^{exp}_t) = \bar{\mathbf{s}} + \mathbf{U}^{id}\mathbf{p}^{id}_t + \mathbf{U}^{exp}\mathbf{p}^{exp}_t. \tag{3.1}$$

Here, $\bar{\mathbf{s}} \in \mathbb{R}^{3N}$ is the mean shape of the 3DMM, given as $\bar{\mathbf{s}} = \bar{\mathbf{s}}^{id} + \bar{\mathbf{s}}^{exp}$, with $\bar{\mathbf{s}}^{id}$ and $\bar{\mathbf{s}}^{exp}$ corresponding to the mean identity shape and mean expression deformation respectively. The orthonormal basis of identity $\mathbf{U}^{id} \in \mathbb{R}^{3N \times n_{id}}$ has n_{id} principal components ($n_{id} \ll 3N$), while $\mathbf{U}_{exp} \in \mathbb{R}^{3N \times n_e}$ corresponds to the orthonormal basis of expression, after keeping n_{exp} principal components ($n_{exp} \ll 3N$). Moreover, $\mathbf{p}^{id}_t$ and $\mathbf{p}^{exp}_t$ denote the parameters of identity and expression to be determined, as they represent the information of the facial shape at frame t. As can be seen in Eq. (3.1), the shape vertices of the face $\mathbf{s}_t$ are computed from the identity and expression parameters. Throughout our experiments, we use the identity 3DMM $\{\bar{\mathbf{s}}^{id}, \mathbf{U}^{\mathbf{id}}\}$ that originates from LSFM [BRP+18], built from more than 10,000 scans of different people, which is one of the largest 3DMMs, with rich demographic variations. The expression 3DMM, denoted as $\{\bar{\mathbf{x}}_{exp}, \mathbf{U}_{exp}\}$ comes from [ZCR+17] and has been built with the blendshapes of Facewarehouse [CWZ+13] by adopting Nonrigid ICP [CMZP17] in order to register the expression model with LSFM.

3.1.2 3DMM Video Fitting

Our objective during the 3D face reconstruction stage is to estimate the per frame facial shape parameters (expression and identity) from a given video, along with the camera parameters. To that end, we adopt a landmark-based approach that considers the information of all video frames concurrently. As proposed in [DRC+19], we exploit the highly-reliable algorithms for landmark localisation and fit the 3DMMs into videos with the assistance of facial landmarks. The adopted 3D face recovery algorithm consists of two steps: an initialisation process that yields the camera parameters and the main fitting step that estimates the expression and identity parameters.

During the initialisation stage, we make the assumption that the facial shape to be recovered remains rigid throughout the video, and seek to estimate the identity $\mathbf{p}^{id}$ and expression parameters $\mathbf{p}^{exp}$ of the rigid facial shape, along with the per-frame camera parameters $\mathbf{p}^c_t$. This assumption might be simplistic but it is adequate for the accurate estimation of camera parameters, since head pose variations can be interpreted as the linear motions of a rigid shape. Camera parameters are estimated by minimising the re-projection error:

$$\min_{\mathbf{p}^c_{1:T}, \mathbf{p}^{id}, \mathbf{p}^{exp}} \mathcal{E}_l(\mathbf{p}^c_{1:T}, \mathbf{p}^{id}, \mathbf{p}^{exp}, \mathbf{l}_{1:T}) = \sum_{t=1}^{T} ||\mathcal{Q}(\mathcal{P}(\mathcal{S}(\mathbf{p}^{id}, \mathbf{p}^{exp}), \mathbf{p}^c_t)) - \mathbf{l}_t||_2^2, \tag{3.2}$$

where $\mathcal{S}$ denotes the shape, as formulated in Eq. (3.1), $\mathcal{P}$ is the projection operation for scaled orthographic projection and $\mathbf{l}_t \in \mathbb{R}^{2L}$ are the given 2D landmarks for frame t (L points in total). Lastly, $\mathcal{Q} : \mathbb{R}^{3N} \rightarrow \mathbb{R}^{2L}$ is the vertex sampling operator, which selects only the vertices that are associated with the L landmarks and drops the depth dimension. We solve the optimisation problem above by applying constraints on the minimum and maximum values of identity and expression parameters and following an alternating minimisation approach. That is, in one step we keep the shape parameters fixed and update camera parameters and in the next step we keep the camera fixed and update the shape. This is a box-constrained optimisation problem that we solve using a least squares method and reflective Newton [CL96]. After that, we keep only the estimated camera parameters $\mathbf{p}^c_{1:T}$ and discard the recovered shape values.

For the main fitting stage, we assume that the identity parameters remain fixed throughout the frames, such that $\mathbf{p}^{id}_t = \mathbf{p}^{id}$, $T = 1, \ldots .T$. However, we let the expression parameters $\mathbf{p}^{exp}_t$ vary from one frame to another. In addition, we utilise the camera parameters recovered in the previous stage. Here, we optimise an error function that is made up of three terms:

$$\min_{\mathbf{p}^{id}, \mathbf{p}^{exp}_{1:T}} \mathcal{E}_{fit}(\mathbf{p}^c_{1:T}, \mathbf{p}^{id}, \mathbf{p}^{exp}_{1:T}, \mathbf{l}_{1:T}) = \mathcal{E}_l(\mathbf{p}^c_{1:T}, \mathbf{p}^{id}, \mathbf{p}^{exp}_{1:T}, \mathbf{l}_{1:T}) + \lambda_p \mathcal{E}_p(\mathbf{p}^{id}, \mathbf{p}^{exp}_{1:T}) + \lambda_s \mathcal{E}_s(\mathbf{p}^{exp}_{1:T}). \quad (3.3)$$

The first term $\mathcal{E}_l$ is a 2D landmark re-projection loss, which is the same error that appears in Eq. (3.2). The second term $\mathcal{E}_p$ enforces a quadratic prior to each expression parameter and the identity parameters and the third term $\mathcal{E}_s$ imposes temporal smoothness over the recovered expression, by computing a quadratic error of expressions second derivatives, with respect to time. The contribution of each term to the total loss is regulated with hyper-parameters λ_p and λ_s. In order to cope with outliers, such as particular frames with occlusions and significant inaccuracies in landmark prediction, we enforce constraints both on identity and expression parameters. This optimisation problem is solved with the reflective Newton method [CL96].

Last but not least, we track gaze, since eye movement is a very important aspect of reenactment. Given that 3DMMs do not contain gaze information, the tracking algorithm described above is not able to capture the direction of eyes. To that end, we use a gaze tracking method similar with the one adopted by [PSH18] to track the eye motion of the source and target videos.

3.1.3 Conditional Face Representation

Considering that we approach the problem of reenactment as a video-to-video translation task, we do not pass the recovered per-frame shape and camera parameters directly into our neural network. Instead, we adopt a more appropriate representation of the 3D facial shape, projected into the 2D image plane, which provides our video rendering network with semantic information of the target face. Given a sequence of 3D facial shapes $\mathbf{s}_{1:T}$, which are computed from the estimated identity and expression parameters with Eq. (3.1), as well as the estimated camera parameters $\mathbf{p}^c_{1:T}$, we rasterise each 3D shapes and compute a visibility mask $\mathbf{M} \in \mathbb{R}^{H \times W}$. Each pixel of the mask keeps the index of the visible triangle in the 3D shape that gets projected into this pixel. Then, we resort to the 3DMM's mean shape $\bar{\mathbf{s}}$ in order to calculate the x-y-z coordinates of each visible triangle, normalised in the range $[0, 1]$. In this way, we get a semantic representation of the face, known as PNCC [ZLL+16] images, where each pixel stores semantic information of the face in the form of 3D coordinates, which are interpreted as colors. In addition to PNCC images, we condition our video rendering network on gaze images. The

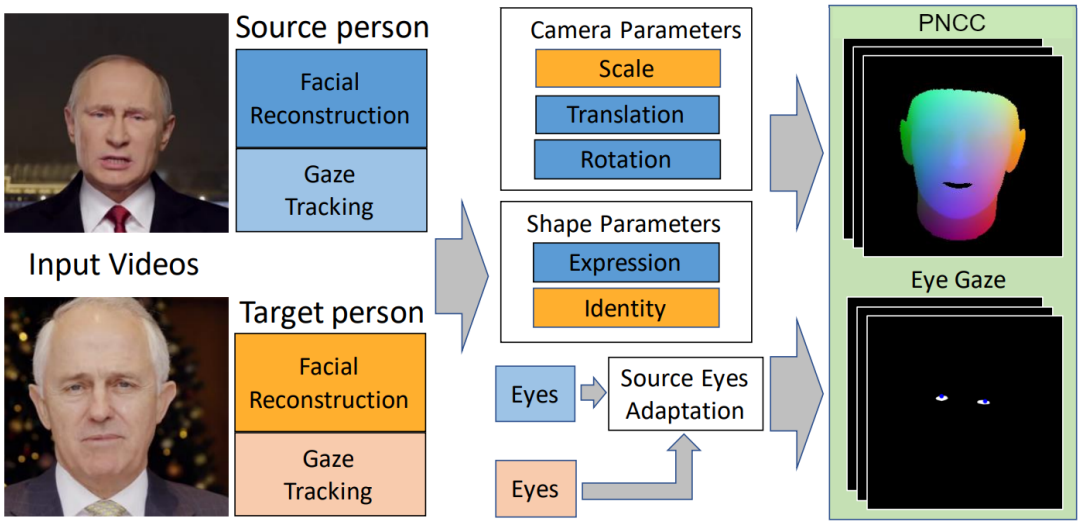


Figure 3.1: Our face tracking pipeline is used to recover the camera and shape parameters of the source and target faces and track their gaze. This stage produces the conditional input for our video rendering network. During inference, we take the expressions as well as the translation and rotation parameters from the source frames.

adopted gaze images are in a "1-1" correspondence with the PNCCs, as in both representations eye regions appear at the same pixel locations. An illustration of PNCCs and gaze images appears in Fig. 3.1, highlighted with green colour. The input to our video rendering network is formed by concatenating PNCC images with eye gaze images channel-wise.

Let $\mathbf{y}_{1:T}$ denote the target video. In order to train a rendering network on this identity, we form a conditional input sequence, by processing the frames of the target video with our tracking system. This process yields the sequence of conditional inputs, denoted as $\mathbf{x}_{1:T} = \{\mathbf{x}_t \in \mathbb{R}^{H\times W\times 6} | t = 1, \dots T\}$. However, during inference, we compute the camera and shape parameters from the both the source and target video. Then, we replace the identity parameters as well as the scale (camera) parameter of the source subject with the parameters coming from the target video, as depicted in Fig. 3.1. This enables to replace the facial geometry of the source with the target one and alleviate the identity preservation problem of reenactment, which is caused by the training-test gap, occurring due to the different facial shape of the source.

3.2 Video Neural Rendering

Given a sequence of conditional inputs $\mathbf{x}_{1:T}$, extracted from any source video, our generative network learns to translate this sequence into to a photo-realistic temporally stable predicted sequence $\tilde{\mathbf{y}}_{1:T}$. This synthetic video depicts the target subject performing the same facial expressions, head movements and changes in gaze with the source actor. We optimise this network using a self-reenactment setup, where the source video is the same with the target one. In this way, the synthetic video $\tilde{\mathbf{y}}_{1:T}$ has to be a reconstruction of the ground video $\mathbf{y}_{1:T}$. For this task, we make use of a GAN framework that performs video-to-video translation and is composed of a generator G that is optimised together with its adversaries: an image discriminator D_I and a dynamics discriminator D_D. The discriminators ensure that the generated video is photo-realistic, temporally stable and follows the facial motions with the real video. Furthermore, we

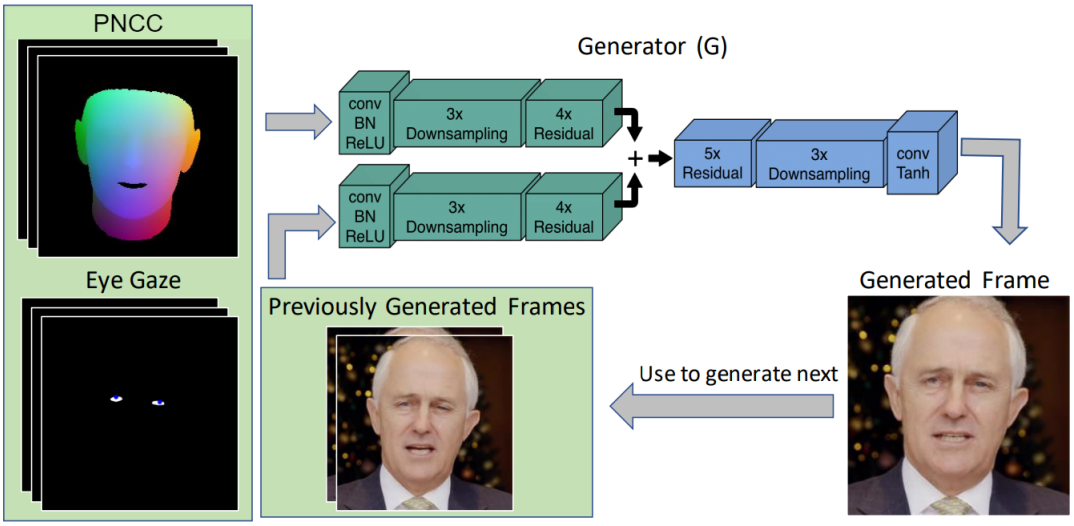


Figure 3.2: The proposed rendering network is based on the architecture of the generator introduced in vid2vid [WLZ+18a]. However, unlike vid2vid, we use a single decoding branch.

enhance the visual quality around the mouth region by adding a mouth discriminator network D_M to the adversarial framework.

3.2.1 Generator

We model the temporal dependence of generated frames on information from past time steps by adopting a sequential model [WLZ+18a] that conditions synthesis of frame t on the conditional inputs $\mathbf{x}_{t-2}$, $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$, but also on the previously generated frames $\tilde{\mathbf{y}}_{t-1}$ and $\tilde{\mathbf{y}}_{t-2}$. Therefore, the synthesised frame t is given as:

$$\tilde{\mathbf{y}}_t = G(\mathbf{x}_{t-2:t}, \tilde{\mathbf{y}}_{t-2:t-1}; \theta_G). \tag{3.4}$$

We apply the generator sequentially and produce the output frames one after the other, until the full sequence has been generated. The architecture of our network is based on the generator of vid2vid [WLZ+18a]. It consists of two identical encoders, with the first one processing the conditional input $\mathbf{x}_{t-2:t}$, and the second one encoding the two former synthetic frames $\tilde{\mathbf{y}}_{t-2:t-1}$. The resulting feature maps are added together and then processed by the decoding layers, which bring the generated frame $\tilde{\mathbf{y}}_t$ in the normalised range $[-1, 1]$, with a tanh function. In Fig. 3.2, we illustrate an overview of the proposed generator.

3.2.2 Discriminators

Image and mouth discriminators, D_I and D_M These two networks learn to distinguish between real and synthetic frames and help the generator produce more realistic samples. During training, we feed the real data $(\mathbf{x}_t, \mathbf{y}_t)$ as well as the synthetic pair $(\mathbf{x}_t, \tilde{\mathbf{y}}_t)$ into network D_I, which predicts a scalar for each pair that corresponds to a real or fake label. Moreover, we use facial landmarks to detect and crop the mouth regions $(\mathbf{x}_t^{mouth}, \mathbf{y}_t^{mouth})$ and $(\mathbf{x}_t^{mouth}, \tilde{\mathbf{y}}_t^{mouth})$, and pass them through the mouth discriminator D_M.

Dynamics discriminator D_D The dynamics discriminator learns to spot frame sequences with implausible temporal dynamics. This discriminator receives as an input $K = 3$ real frames $\mathbf{y}_{t-K+1:t}$ or generated frames $\tilde{\mathbf{y}}_{t-K+1:t}$ along with the optical flow field $\mathbf{w}_{t-K+2:t}$, which has been estimated from the ground truth frames before training the networks. The goal of discriminator D_D is to make sure that the flow $\mathbf{w}_{t-K+2:t}$ agrees with the motion in the given frames. To that end, the dynamics discriminator is trained to classify $(\mathbf{w}_{t-K+2:t}, \mathbf{y}_{t-K+1:t})$ as real and $(\mathbf{w}_{t-K+2:t}, \tilde{\mathbf{y}}_{t-K+1:t})$ as fake. In reality we use a dynamics discriminator that detects unrealistic facial motions at three different temporal scales. At the first scale, network D_D processes consecutive frames. At the two other scales we do not choose subsequent frames, as we sub-sample frames for each scale, each time by a factor of K. That is, the frames at scale number two have a distance of 3 time steps, while at the third scale frames have a distance of 9 time steps. This multi-scale approach, which was adopted originally in [WLZ+18a], helps to achieve short and longer term temporal consistency.

In order to help the generator produce high-frequency details in local patches of images we adopt PatchGAN [IZZE17], [WLZ+18a] in all discriminators. Moreover, all discriminators have the same architecture, borrowed from the critic of pip2pixHD [WLZ+18b].

3.2.3 Facial Optical Flow

In order to produce as realistic motions as possible with our rendering network, it is important to condition network D_D on a very accurate optical flow of the target face. Many optical flow estimation methods perform flow prediction without making any presumptions about the objects that appear in frames. Given that faces perform non-linear deformations due to the complex interactions of multiple facial muscles, it is obvious that capturing the facial optical flow with the assistance of one off-the-shelf method for optical flow prediction might not lead to very accurate results. We overcome this problem by capitalising on previous knowledge on video portraits and training a flow network specialised on faces. To that end, we employ the architecture of FlowNet 2.0 [IMS+17] for the estimation of optical flow. We start from a pre-trained checkpoint of [IMS+17] and fine-tune FlowNet 2.0 on 4DFAB [CKPZ18] database, which consists of 4D high-resolution videos of subjects performing various actions. For the facial region we prepare a pseudo-ground truth 2D optical flow, using the 3D scans of 750K frames. For that, we compute the difference between consecutive meshes and project it into the image plane to represent the 2D flow. For the background region, we employ a masked end-point-error loss. In this way, the predicted flow in the background remains the same with the original FlowNet 2.0 prediction, while the foreground (facial area) flow is supervised by the ground truth 2D optical flow that comes from 4DFAB.

3.2.4 Objective functions and Optimisation

We optimise the rendering network in a supervised way, on the task of self-reenactment, with the assistance of an an adversarial loss. With LSGAN [MLX+17], the adversarial loss for training the generator is:

$$\mathcal{L}_{adv}^{G} = \mathbb{E}_t\Big[\big(D_I(\mathbf{x}_t, \tilde{\mathbf{y}}_t)-1\big)^2+\big(D_M(\mathbf{x}_t^{mouth}, \tilde{\mathbf{y}}_t^{mouth})-1\big)^2+\big(D_D(\mathbf{w}_{t-K+2:t}, \tilde{\mathbf{y}}_{t-K+1:t})-1\big)^2\Big]. \quad (3.5)$$

We further include two reconstruction loss terms to the objective of the rendering network: 1) a perceptual loss $\mathcal{L}^G_{vgg}$ and 2) a feature matching loss $\mathcal{L}^G_{feat}$ [WLZ+18b] that is computed based on the image discriminators. Let $\mathbf{y}_t$ be the ground frame and $\tilde{\mathbf{y}}_t$ be the generated one. Then, we extract visual feature maps from both images, using different layers of a VGG network [SZ14], which is pre-trained on ImageNet dataset [DDS+09]. The perceptual loss term for frame t is:

$$\mathcal{L}^G_{vgg} = \sum_i \frac{1}{H^{(i)}_L W^{(i)}_L C^{(i)}_L} ||L^{(i)}(\mathbf{y}_t) - L^{(i)}(\tilde{\mathbf{y}}_t)||_1. \tag{3.6}$$

$L^{(i)}$ denotes the activation of VGG network's i-th layer, a feature map of size $H^{(i)}_L \times W^{(i)}_L \times C^{(i)}_L$.

In a similar way, we compute the feature matching loss. We extract visual features with the image and dynamics discriminators D_I and D_D and then compute the ℓ_1-distance of the feature maps coming from the synthetic images $\tilde{\mathbf{y}}_t$ and the corresponding ground image $\mathbf{y}_t$:

$$\begin{aligned}\mathcal{L}^G_{feat} = &\frac{1}{2}\sum_j \frac{1}{H^{(j)}_{D_I} W^{(j)}_{D_I} C^{(j)}_{D_I}} ||D^{(j)}_I(\mathbf{x}_t, \mathbf{y}_t) - D^{(j)}_I(\mathbf{x}_t, \tilde{\mathbf{y}}_t)||_1 \\ &+ \frac{1}{2}\sum_k \frac{1}{H^{(k)}_{D_M} W^{(k)}_{D_M} C^{(k)}_{D_M}} ||D^{(k)}_M(\mathbf{x}^{mouth}_t, \mathbf{y}^{mouth}_t) - D^{(k)}_M(\mathbf{x}^{mouth}_t, \tilde{\mathbf{y}}^{mouth}_t)||_1\end{aligned} \tag{3.7}$$

The overall objective function for network G is:

$$\mathcal{L}^G = \mathcal{L}^G_{adv} + \lambda_{vgg}\mathcal{L}^G_{vgg} + \lambda_{feat}\mathcal{L}^G_{feat} \tag{3.8}$$

For our experiments, we set $\lambda_{vgg} = \lambda_{feat} = 10$.

The discriminators are trained using their adversarial objective functions. To be more specific, the image discriminator is optimised by minimising the following term:

$$\mathcal{L}^{D_I}_{adv} = \frac{1}{2}\mathbb{E}_t\Big[\big(D_I(\mathbf{x}_t, \mathbf{y}_t) - 1\big)^2\Big] + \frac{1}{2}\mathbb{E}_t\Big[\big(D_I(\mathbf{x}_t, \tilde{\mathbf{y}}_t)\big)^2\Big]. \tag{3.9}$$

The dedicated mouth discriminator D_M is optimised with a similar adversarial term with $\mathcal{L}^{D_I}_{adv}$, which results by replacing the images in Eq. (3.9) with the cropped mouth areas:

$$\mathcal{L}^{D_M}_{adv} = \frac{1}{2}\mathbb{E}_t\Big[\big(D_I(\mathbf{x}^{mouth}_t, \mathbf{y}^{mouth}_t) - 1\big)^2\Big] + \frac{1}{2}\mathbb{E}_t\Big[\big(D_I(\mathbf{x}^{mouth}_t, \tilde{\mathbf{y}}^{mouth}_t)\big)^2\Big]. \tag{3.10}$$

Finally, the dynamics discriminator is trained by minimising the adversarial loss:

$$\mathcal{L}^{D_D}_{adv} = \frac{1}{2}\mathbb{E}_t\Big[\big(D_D(\mathbf{w}_{t-K+2:t}, \mathbf{y}_{t-K+1:t}) - 1\big)^2\Big] + \frac{1}{2}\mathbb{E}_t\Big[\big(D_D(\mathbf{w}_{t-K+2:t}, \tilde{\mathbf{y}}_{t-K+1:t})\big)^2\Big]. \tag{3.11}$$

3.3 Experimental Analysis

In this section, we study the generative abilities of our proposed model when it comes to transferring the pose, eye gaze and facial expression from a source video to a target actor. We compare our approach with related methods and show that it achieves very competitive results



Figure 3.3: Examples of frames from the videos we used throughout our experiments.

in terms of photo-realism. We perform numerous quantitative and qualitative experiments, including an ablation study, in order to evaluate the performance of our proposed methodology. To that end, we formed a small dataset with video portraits, all at the same 256^2 resolution. For that, we collected six videos of well-known politicians from YouTube, under the Creative Commons (CC) licence. Furthermore, the authors of [KGT+18] kindly accepted to share with us three videos from their database along with their results, which have been included to our experiments. Fig. 3.3 shows one representative frame from each video in our dataset. Please note that only one identity appears per video.

Our implementation of Head2Head GAN framework requires a few minutes of video footage for training a person-specific model on a given subject. The entire training procedure takes around five hours on a single NVIDIA Titan V GPU. We use Adam solver [KB14] with learning rate $\eta = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and a batch size of one.

3.3.1 Quantitative Results

We assess our method's performance quantitatively, by carrying out two types of experiments: a self-reenactment and a cycle-reenactment experiment. During self-reenactment, we split every video into a training and a test split. We use the first two thirds for training and the last one third for testing. Then, we calculate the mean per-pixel error on the test set. Please note that we define the aforementioned error as the ℓ_2-distance between synthetic and ground truth the pixels of test frames. We assume that value of each channel is in the range $[0, 255]$. In Fig. 3.4, we demonstrate the quantitative results we obtained on four videos-individuals (Trudeau, Biden, Merkel and Turnbull). Each video consists of around four thousand training frames.

For our next numerical experiment, we performed cycle-reenactment in order to evaluate the performance of Head2Head while transferring the expression, pose and gaze from a source video to a target and then back to the original source. To be more specific, given a test source video $\mathbf{y}_{src}^{(test)}$ and a training target video $\mathbf{y}_{tgt}^{(train)}$, we first optimise a generator G_{tgt} using the target video. After that, we perform inference and transfer the facial motions from the test source footage to the target identity, which yields $\tilde{\mathbf{y}}_{tgt} = G_{tgt}(\mathbf{x}_{src}^{(test)})$. We close the cycle by training

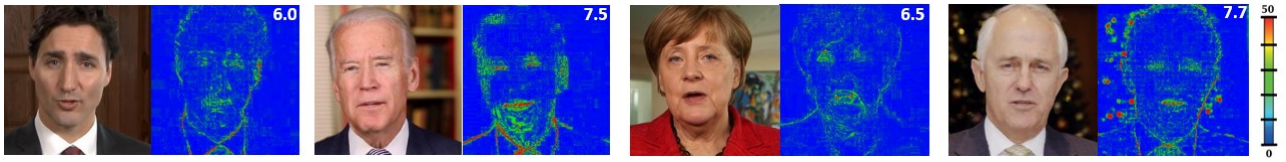


Figure 3.4: Quantitative results for the self-reenactment task, using four identities. For each target identity we show a synthesised frame example and he corresponding error map with respect to the the corresponding real frame. In the top-right corners of heatmaps we report the average reconstruction error, across the entire test sequence.

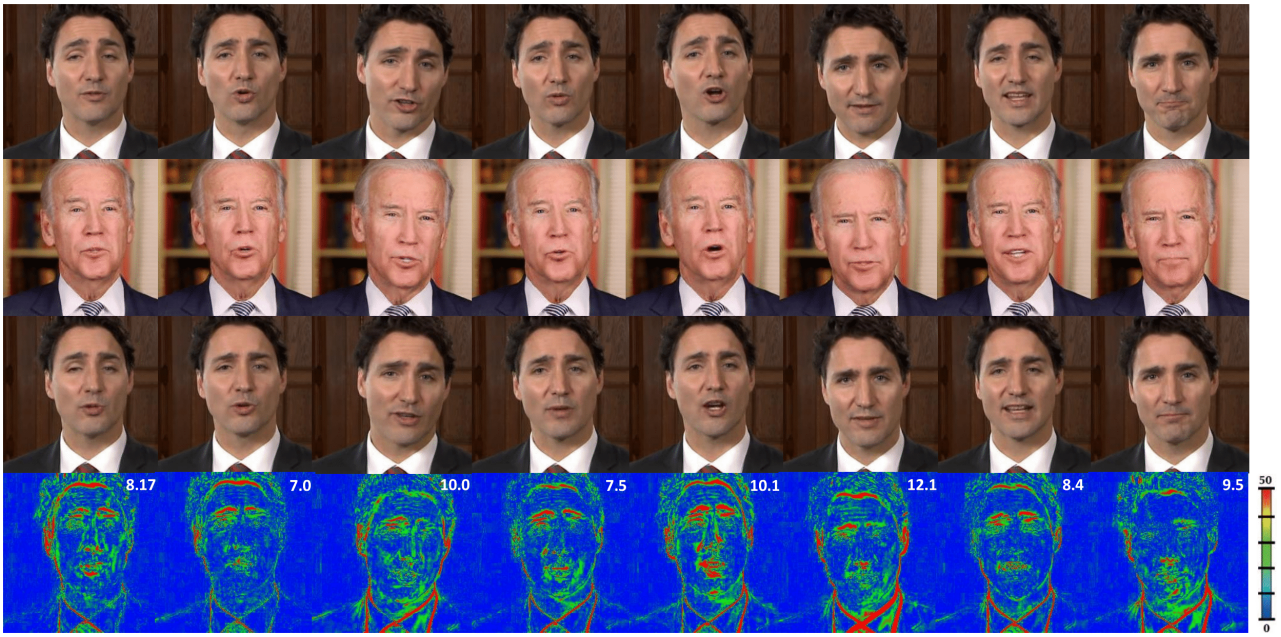
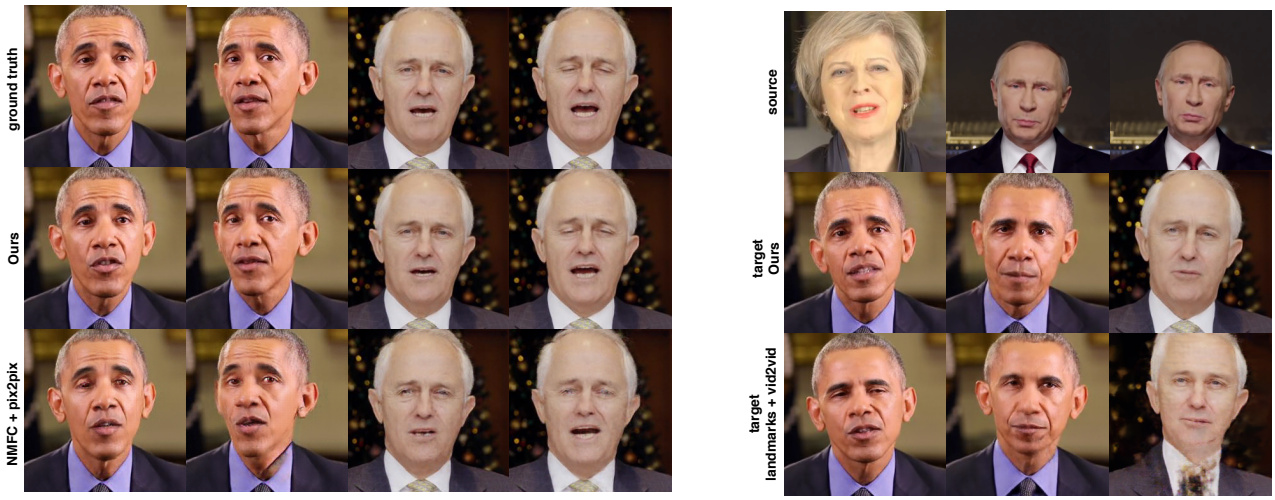


Figure 3.5: Cycle-reenactment results of our method. From first to last row: source frames, synthetic frames of the target subject, synthetic frames of the source identity driven by the synthetic frames of the target identity, heatmaps with per-pixel l_2-distance between the ground and generated source frames. In the top-right corners of heatmaps we present the mean per-pixel distance for each frame.

a second generator G_{src} on the source actor using a training video $\mathbf{y}_{src}^{(train)}$ that has different frames from $\mathbf{y}_{src}^{(test)}$. Then, we use the generated video $\tilde{\mathbf{y}}_{tgt}$ to drive network G_{src} and synthesise $\tilde{\mathbf{y}}_{src} = G_{src}(\tilde{\mathbf{x}}_{tgt})$. After that, we measure the mean per-pixel error between sequences $\tilde{\mathbf{y}}_{src}$ and $\mathbf{y}_{src}^{(test)}$. In Fig. 3.5, we demonstrate various examples of source, synthetic target and synthetic source frames, along with the emerging error heatmaps, between the ground source and synthetic source frames. The mean per-pixel distance across one thousand test frames is 9.3.

3.3.2 Qualitative Evaluation

Here, we demonstrate the visual quality and realism that can be achieved with Head2Head system. We further compare our approach with Deep Video Portraits (DVP) [KGT+18], a state-of-the-art system for full head reenactment, as well as with two strong baselines: pix2pix conditioned on PNCCs and eye gaze images [IZZE17] and vid2vid conditioned on sketches drawn



(a) (b)

Figure 3.6: (a) Visual comparison of the proposed Head2Head with pix2pix baseline [IZZE17], conditioned on PNCC images, for the task of self-reenactment. As can be seen, we achieve superior image quality and more accurate reconstruction. (b) Qualitative comparison of Head2Head with vid2vid [WLZ+18a] conditioned on landmark sketches. The adoption of a 3DMM in conjunction with PNCC inputs achieves identity and expression disentanglement, as our rendering network preserves the characteristics of the target, while vid2vid distorts them.

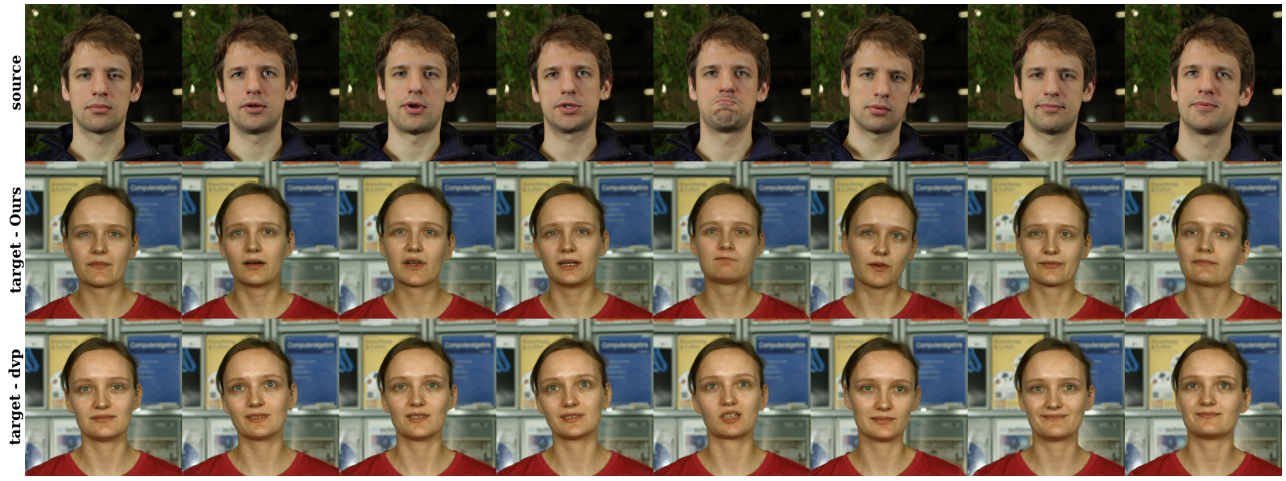


Figure 3.7: Comparison with DVP [KGT+18]. From top to bottom rows: source frames, target frames produced by Head2Head, the same frames generated with DVP. Our generated head poses and facial expressions follow the source performances closer than DVP.

from sparse 2D landmarks [WLZ+18a]. We optimised each of the two baseline architectures and our model on two target videos-identities. After that, we carried out a self-reenactment experiment for both target videos. Our findings indicate that the absence of sequential modeling in image-to-image translation methods such as pix2pix results in frames with artifacts and temporal inconsistencies. Examples of frames from the self-reenactment experiment are displayed in Fig 3.6 (a). As can be seen, on the task of video reconstruction, Head2Head generates more plausible results when compared to pix2pix model, even though we drive pix2pix also with PNCC and gaze images.

Next, we demonstrate the importance of adopting a parametric 3D face representation along with a PNCC conditional input in cross-identity reenactment. Given any source video portrait, our proposed method is able to disentangle the identity-related facial shape from expression-related deformations, and at the same time preserve the facial traits of the target subject. On the contrary, vid2vid conditioned on landmark sketches [WLZ+18a] appears to distort the shape of the target face, since the sparse 2D landmarks pass identity-related information from the source frames to the generated video of the target, as depicted in Fig. 3.6 (b). Please note that in order to make a fair comparison, we resized the facial landmarks of the source to the target head size. In this way we minimise the artifacts and distortions that come from the different head scales, or else vid2vid outputs non-realistic images with even more artifacts, due to the training-test gap.

Lastly, when compared to DVP [KGT+18], Head2Head generates image samples of equivalent photo-realism and at the same time achieves more convincing results in the mouth region. Moreover, as can be observed in Fig. 3.7, there are cases where our method performs significantly better than DVP [KGT+18] when it comes to pose and expression transfer. The head poses and expressions of our synthetic samples follow very closely the ones in the source frames.

3.3.3 Ablation Study

We further evaluate our design choices and carry out an ablation study that demonstrates the importance of our system's key components. First, we show the shortcomings of using the optical flow estimated by FlowNet 2.0 [IMS+17] instead of our facial optical flow. Table 3.1 (a) shows the mean errors we recorded when performing self-reenactment experiment. As can be seen, we achieve smaller errors in all video portraits when using facial flow. This is expected, as the flow estimator we use was fine-tuned exclusively on facial videos. In addition, we explore the importance of the dedicated mouth discriminator D_M under self-reenactment with four different individuals. For that, we measure the mean pixel distance between the generated and the ground truth frames, for the small area around the mouth. Synthesising the interior of the mouth is a difficult task, which includes high errors due to the absence of conditional information for the tongue and teeth. The average pixel distances we report in Table 3.1 (b) show that the incorporation of a mouth discriminator notably improves our samples. The improvements are also visualised in Fig. 3.8.

Table 3.1: Numerical results of the ablation study.

(a) The mean pixel error on the task of self-reenactment with the identities show in Fig. 3.4, when using optical flows either from FlowNet 2.0 [IMS+17] or our facial flow.

Video	Head2Head w/ FlowNet 2.0 (↓)	Head2Head w/ Facial Flow (↓)
Justin	6.9	**6**
Joe	9.3	**7.7**
Merkel	9	**7.5**
Turnbull	9.4	**6.5**

(b) The mean pixel error on the task of self-reenactment, for the image patch around the mouth, with and without the mouth discriminator D_M.

Video	Head2Head w/o D_m (↓)	Head2Head w/ D_m (↓)
May	11.2	**10.1**
Justin	11	**10.4**
Turnbull	11.3	**10.7**
Obama	22.3	**19.4**

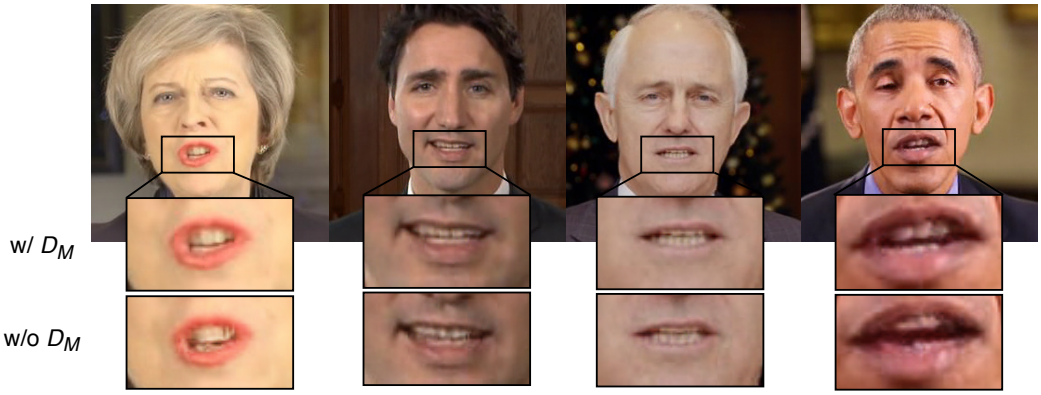


Figure 3.8: Examples of frames synthesised by Head2Head, with and without the dedicated mouth discriminator D_M. It is evident that network D_M improves the quality of teeth.

3.3.4 Automated Study

Recently, there has been a plethora of attempts to detect synthetic or manipulated videos [RCV+19] automatically. For instance, the convolutional neural network presented in [RCV+19] manages to outperform human observers when it comes to detecting manipulated video data. The dataset they used to train the forgery detection network is among the largest available datasets with synthetic faces. It was created by manipulating around 1,000 YouTube videos, with graphics-based [TZS+16, Mar19] and learning-based face reenactment methods [tor19, TZN19]. In total, their network was trained on 1.8 million synthetic images and achieved a high forgery detection accuracy on the test split, around 99%.

We use the pre-trained network provided by the authors of [RCV+19] in order to evaluate the quality of our generated samples automatically. To that end, we selected fifty random videos from their database and performed self-reenactment with Head2head. Then we used their trained forgery detection network and measured its accuracy on our synthetic data. We recorded a accuracy of only 51.88%. These results show the high photo-realism of our generated samples, since even a well-trained forgery detection network struggles to distinguish them from real frames.

3.3.5 User Study

Furthermore, we conducted a user study that consists of two parts. First, we presented real and synthetic videos from Head2Head to 95 anonymous participants. We asked them to rate their photo-realism. Our synthetic videos were produced from three different reenactment tasks: full head reenactment, face reenactment and self-reenactment. We presented each participant with 5-second videos, and we asked them the question: *"On a scale of 1-5, how real does this video look?"*. We explained that 1 corresponds to "absolutely fake" and 5 to "definitely real". The ratings we collected from the 95 participants are displayed in Table 3.2 and Table 3.3. We consider videos that received 4 or 5 as "real". According to the ratings, the percentage of synthetic videos that were rated as "real" is comparable to the percentage of real videos that were also rated as "real". This suggests that Head2Head creates video portraits almost indistinguishable from real-world data. Please note that the participants were very cautious about finding the synthesised videos, since the task had been explained to them in advance.

Table 3.2: The ratings of 95 participants on the task of self-reenactment. In columns numbered from 1 to 5 we show the number of people that gave that rating. The next column, named "real", displays the percentage of participants that rated the video with a 4 or a 5.

Video	**Our synthetic videos**						**Real videos**					
	1	2	3	4	5	"real" (↑)	1	2	3	4	5	"real" (↑)
May	5	8	12	34	36	**74%**	3	6	6	26	54	84%
Turnbull	11	15	20	29	20	**52%**	2	6	14	40	33	77%
Putin	4	10	18	36	27	**66%**	2	3	13	37	40	81%
Justin	5	8	18	36	28	**67%**	5	7	11	28	44	76%
Target from Fig. 3.7	3	11	14	26	41	**71%**	9	12	21	29	24	56%
Mean scores	6	10	16	32	30	**66%**	4	7	13	32	39	**75%**

Table 3.3: The ratings of 95 participants on the task of face reenactment and full head reenactment.

Video	**Our synthetic videos**						**Real videos**					
	1	2	3	4	5	"real" (↑)	1	2	3	4	5	"real" (↑)
Merkel (face reenactment)	16	21	20	22	16	**40%**	1	6	17	31	40	**75%**
Justin (full head reenactment)	8	12	25	39	11	**53%**	5	7	11	28	44	76%
Target from Fig. 3.7 (full head reenactment)	37	19	10	22	7	**31%**	9	12	21	29	24	56%
Mean scores (full head reenactment)	22	16	18	30	9	**42%**	7	10	16	28	34	**66%**

Finally, we showed the participants a source video and two synthesised videos, one produced by Head2Head and another one synthesised by DVP [KGT+18], with the source and target identities displayed in Fig. 3.7. Both synthetic sequences were generated from a full head reenactment experiment. After that, we posed the following question: *"Which video follows the movements of the person in the source better?"*. 70 participants answered the question and 42 of them (60%) chose the video generated by Head2Head, which reveals that our proposed method managed to reenact the source performance more successfully than DVP method.

3.4 Multiperson Synthesis with Few-shot Learning

Despite the high quality of generated video samples, the Head2Head method described above is a strictly person-specific method. It requires a few thousands of frames of the target identity and needs to optimise a separate set of networks (generator and discriminators) for every new target video and identity. In this section, we propose an extension of Head2Head that enables to synthesise multiple identities. For that cause, we introduce a new module to our framework, which computes identity embeddings from image portraits. Furthermore, we modify the architecture of the generator and image discriminator to accommodate multiperson synthesis. During training, we adopt a few-shot learning strategy, similar with the one proposed in [ZSBL19] that consists of two stages. First, we optimise our GAN-based framework on a video database of many different identities. After that, we fine-tune the model on a single short video of an unseen target subject. In the following, we describe the new architecture and training algorithm in detail. Then, we report the experimental findings that demonstrate the merits of extending our method towards the direction of few-shot learning.

3.4.1 Initial Training Stage

For the initial learning stage of our extended rendering network, we utilise a dataset with multiple identities. Let N be the number of videos-identities and $\mathbf{y}^{(i)}_{1:T_i} = \{\mathbf{y}^{(i)}_t | t = 1, \ldots, T_i\}$ denote the video of the i-th identity in our training database, which has length T_i. Please note that we have removed the background of videos. Let $\mathbf{m}^{(i)}_{1:T_i}$ be the corresponding sequence of foreground masks that have been extracted with a face segmentation network. Moreover, we employ the face tracking method presented in section 3.1 to extract the corresponding PNCC sequence $\mathbf{x}^{(i)}_{1:T_i}$, for every video-identity $i = 1, \ldots, N$. After that, we train our generator G to translate each PNCC sequence into a realistic video of the corresponding target identity $\tilde{\mathbf{y}}^{(i)}_{1:T_i}$. Along with the synthetic frames, network G predicts a foreground mask $\tilde{\mathbf{m}}^{(i)}_{1:T_i}$. We optimise the rendering network in a self-reenactment scenario, where the video sequence $\mathbf{y}^{(i)}_{1:T_i}$ serves as ground truth for the predicted frames.

Except for PNCC images that encode facial expression and head pose, here we condition synthesis also on information about the appearance of the target identity i. In spite of the fact that PNCC sequences contain some identity-related information, that is the shape of the subject's face reflected in the rendered 3D mesh, they do not actually carry any knowledge related to appearance features, like cloth details, hair or skin texture. To that end, we can encode the appearance information of various identities with the assistance of an identity encoding network, denoted as E_{id}. This encoder learns to estimate an appearance-related feature vector $\mathbf{h}^{(i)} \in \mathbb{R}^{n_f}$ from subject i. We use this appearance embedding to guide the generative process that outputs the synthetic frames of target identity i. Following Head2Head method, our extended framework is made up of a dynamics discriminator that makes sure that the generated video is temporally coherent, as well as an image and mouth discriminator, which ensure that the synthesised frames looks photo-realistic. Next, we describe how the identity embedding network and the modified generator operate within the extended Head2Head framework.

Identity Embedder E_{id} Let $\mathbf{y}^{(i)}_{1:T_i}$ be a video in the training corpus. We select M random frames $\mathbf{y}^{(i)}_{m_1}, \ldots, \mathbf{y}^{(i)}_{m_M}$ and pass them through the identity embedder. Lastly, we calculate the average appearance feature vector:

$$\mathbf{h}^{(i)} = \frac{1}{M} \sum_{j=1}^{M} E_{id}(\mathbf{y}^{(i)}_{m_j}; \theta_{E_{id}}) \tag{3.12}$$

We choose M random frames from the training video i instead of one, as by averaging the embedding vectors makes the features $\mathbf{h}^{(i)}$ less dependent on the person's expression and pose that appears in the input frames.

Generator G The modified network G conditions frame synthesis on two distinct types of information: $\mathbf{x}^{(i)}_{1:T_i}$, a representation of head pose and expression (PNCC images) and the target identity's appearance representation $\mathbf{h}^{(i)}$. As in the person-specific setting, we propose a sequential generator that estimates synthetic frames but also foreground masks based on the previous synthesised frames. To be more specific, the generation of frame $\tilde{\mathbf{y}}^{(i)}_t$ and mask $\tilde{\mathbf{m}}^{(i)}_t$ depends on the two previously generated frames $\tilde{\mathbf{y}}^{(i)}_{t-2:t-1}$, that is:

$$\tilde{\mathbf{y}}^{(i)}_t, \tilde{\mathbf{m}}^{(i)}_t = G(\mathbf{x}^{(i)}_{t-2:t}, \tilde{\mathbf{y}}^{(i)}_{t-2:t-1}, \mathbf{h}^{(i)}; \theta_G). \tag{3.13}$$

Here, $\mathbf{x}^{(i)}_{t-2:t} = \{\mathbf{x}^{(i)}_{t-2}, \mathbf{x}^{(i)}_{t-1}, \mathbf{x}^{(i)}_{t}\}$ denotes the conditional PNCC input that corresponds to frames $t-2$, $t-1$ and t, while $\mathbf{h}^{(i)}$ denotes the appearance embedding. We use the same embedding to produce all frames in the sequence. As in the original Head2Head method, the generator is made up of two convolutional encoders and a single decoder. The first encoder receives the PNCC input $\mathbf{x}^{(i)}_{t-2:t}$ and the second one processes the previous synthetic frames $\tilde{\mathbf{y}}^{(i)}_{t-2:t-1}$. After that, we add the resulting feature maps and send them to the decoding block, which predicts the foreground mask $\tilde{\mathbf{m}}^{(i)}_{t}$ and frame $\tilde{\mathbf{y}}^{(i)}_{t}$. We inject the appearance embedding $\mathbf{h}^{(i)}$ into network G through its adaptive normalisation layers, as a modulation input. To that end, we use AdaIN layers [HB17], as the ones presented in [ZSBL19]. Nonetheless, we apply adaptive instance normalisation to all layers of network G, during encoding and decoding. Let $\mathbf{a}$ be the input tensor to an arbitrary normalisation layer in G, with C channels, height H and width W. Then, the modulated output of the normalisation layer at site $c \in C, y \in H, x \in W$ is

$$a'_{cyx} = \gamma_c \frac{a_{cyx} - \mu_c}{\sigma_c} + \beta_c \tag{3.14}$$

where a_{cyx} is the input value at the site before normalization, γ_c is the value of the scaling modulation vector $\boldsymbol{\gamma} \in \mathbb{R}^C$ for channel c and β_c is corresponding value of the modulation vector $\boldsymbol{\beta} \in \mathbb{R}^C$ that represents bias. The parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ depend on the appearance of the target subject that is encoded within the identity embedding $\mathbf{h}^{(i)}$, and given by:

$$\boldsymbol{\gamma} = \mathbf{P}_{\gamma}\mathbf{h}^{(i)}, \tag{3.15}$$

$$\boldsymbol{\beta} = \mathbf{P}_{\beta}\mathbf{h}^{(i)}, \tag{3.16}$$

where $\mathbf{P}_{\gamma} \in \mathbb{R}^{C \times n_f}$ and $\mathbf{P}_{\beta} \in \mathbb{R}^{C \times n_f}$ are the learnable affine parameters of each AdaIN layer, which translate the identity embedding $\mathbf{h}^{(i)}$ to the modulation parameters of the normalization layer. Moreover, μ_c and σ_c are the mean and standard deviation of tensor $\mathbf{a}$ in channel c:

$$\mu_c = \frac{1}{HW}\sum_{yx} a_{cyx}, \tag{3.17}$$

$$\sigma_c = \sqrt{\frac{1}{HW}\sum_{yx}\Big((a_{cyx})^2 - (\mu_c)^2\Big)}. \tag{3.18}$$

Image Discriminator D_I The main functionality of this network is to force the generator to produce more photo-realistic frames, as it learns to distinguish real from synthetic data. The image discriminator D_I takes as input either a real $(\mathbf{x}^{(i)}_t, \mathbf{y}^{(i)}_t)$ and outputs a vector $\mathbf{d}$, instead of a scalar. Similarly to [ZSBL19], the image discriminator stores a learnable set of parameters $\mathbf{W} \in \mathbb{R}^{N \times n_f}$, with every row corresponding to a separate video-identity. Using the video index i, each time we select the correct row $\mathbf{w}_i$. Then, we calculate the score that represents the realism of the given input, as:

$$r = \mathbf{d}^{\top}(\mathbf{w}_i + \mathbf{w}_0) + c. \tag{3.19}$$

$\mathbf{w}_0$ and c are also learnable parameters. Since each time we select the row vector that corresponds to video-identity i, during training each row vector $\mathbf{w_i}$ obtains person-specific values, which resemble the purpose of the identity embeddings produced by network E_{id}.

The mouth and dynamics discriminators are the same with the ones described in section 3.2. In Fig. 3.9 we demonstrate an overview of the entire framework, for the initial training stage.

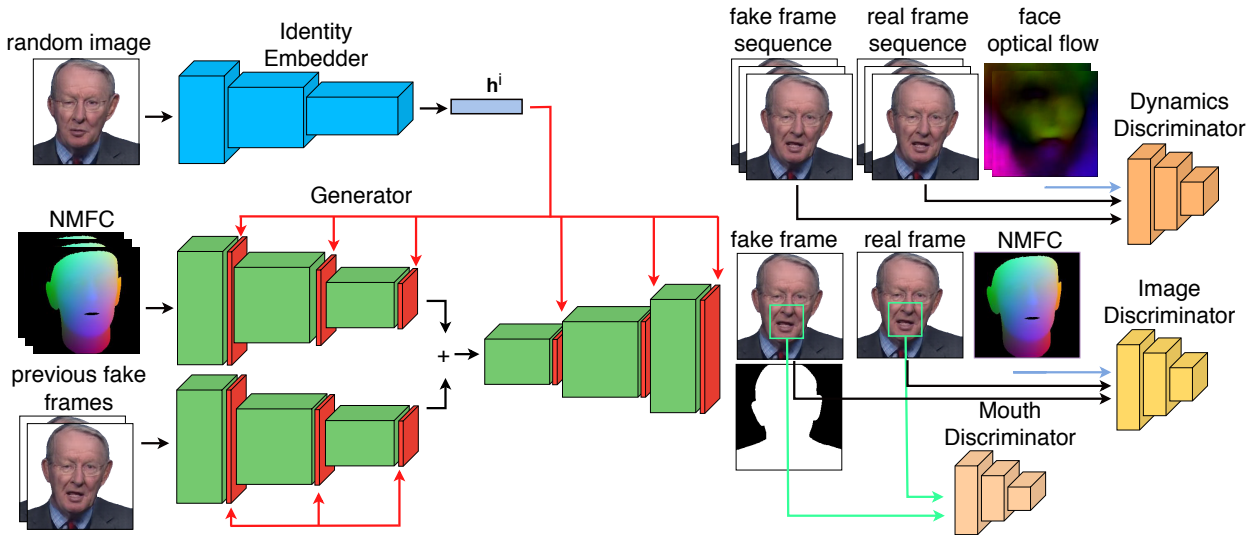


Figure 3.9: The extended Head2Head framework in the initial training stage. The identity embedder estimates the appearance feature $\mathbf{h}^{(\mathbf{i})}$, which is afterwards injected into network G using AdaIN layers.

Optimisation Objective We use the same objective functions with Head2Head, while the parameters of network E_{id} are optimised under the same loss function with the generator network. Nonetheless, following practices from [ZSBL19], we include an extra loss term to the objective of network E_{id}. This loss aims to match the appearance feature vector estimated by the identity embedding network $\mathbf{h}^{(i)}$ with the row of matrix $\mathbf{W}$ that corresponds to video-identity i, that is $\mathbf{w}_i$. To that end, we compute the similarity between the two vectors, as

$$\mathcal{L}_{mch} = 1 - \frac{\mathbf{h}^{(i)^\top}\mathbf{w}_i}{||\mathbf{h}^{(i)}||_2||\mathbf{w}_i||_2}, \tag{3.20}$$

and add this term to the objective of network E_{id} with a hyper-parameter (weight) $\lambda_{mch} = 10$.

3.4.2 Fine-tuning on Unseen Identities

As a second learning step, we fine-tune the components of the proposed GAN using the video footage of an unseen identity. We initialise the learnable parameters with the values we obtained in the first training step. The network initialisation is very important for to obtaining a robust generator, rather than training networks from scratch. Given a short frame sequence $\mathbf{y}^{new}_{1:T}$ and the corresponding foreground masks $\mathbf{m}^{new}_{1:T}$, first we use the face tracker to prepare the PNCC input $\mathbf{x}^{new}_{1:T}$. After that, we send all training frames to the identity embedder, which computes the average appearance embedding

$$\mathbf{h}^{new} = \frac{1}{T}\sum_{j=1}^{T} E_{id}(\mathbf{y}^{new}_j). \tag{3.21}$$

This vector captures the appearance of the new target identity. Then, the embedding network is no longer necessary, as it is not included any further in the training process.

The identity embedding described above is utilised for the initialisation of the AdaIN layers of network G. Then, all AdaIN layers are replaced by ordinary Instance Normalisation (IN) layers [HB17]. For that, first we multiply the optimised projection matrices $\mathbf{P}_\gamma$ and $\mathbf{P}_\beta$ of each adaptive layer with the appearance embedding $\mathbf{h}^{new}$. The results $\boldsymbol{\gamma} = \mathbf{P}_\gamma \mathbf{h}^{new}$ and $\boldsymbol{\beta} = \mathbf{P}_\beta \mathbf{h}^{new}$ serve as the initialisation values of the learnable parameters in instance normalisation layers. The remaining parameters of the generator are initialised with the values learned during the initial learning stage. Regarding the image discriminator, the learnable matrix $\mathbf{W}$ is now replaced by a vector $\mathbf{w}$, which gets initialised with the embedding $\mathbf{h}^{new}$. The rest parameters of network D_I, as well as the mouth and dynamics discriminators D_M and D_V are initialised with the values learned during the initial learning stage. Having all learnable parameters of our framework initialised, we then fine-tune the generator and discriminators on the new target video, following the optimisation process of the original Head2Head.

3.4.3 Experiments

Multiperson Database The dataset we used to train and test the proposed Head2Head extension is made up of 500 videos-identities. Each video consists of one thousand frames. We split the database into two sets of videos. We used the majority of video-identities, around 480, in the initial training step, while the remaining 20 were used in the second fine-tuning stage. The resolution of all frames is 256^2. Each sequence was divided into a training and a testing sub-sequence. We kept the last one hundred frames for testing. In our experiments we fine-tuned the proposed model with sequences of various lengths, from 50 to 500 frames. Our results demonstrate that the generator learns to synthesise convincing samples with as little as 50 training frames. As a pre-processing step, we used a face segmentation network [1] to prepare the foreground masks that serve as ground truth. We also removed the background areas within all frames. In this way, the embedder is not distracted from the background pixels and learns to extract only identity and appearance-related features from images. Since network G learns to estimate the masks, we can then set a background of our choice during inference.

Implementation details The identity embedder borrows its architecture from FaceNet [SKP15] with the embedding vector having a size of $n_f = 512$ values. We optimise all networks with Adam [KB14] solver. We run the initial learning stage for 20 epochs and the second fine-tuning step for around 15 epochs. The fine-tuning step lasts from 3 to 30 minutes per identity, which is determined by the number of training frames (50 to 500). We performed all experiments on a single NVIDIA GeForce RTX 2080 Ti and used a training batch size of one.

Comparison with other systems We pit the generative performance of our approach against a person-agnostic method that is based on 2D warping, namely X2Face [WKZ18] and a video-to-video translation system, vid2vid [WLZ+18a]. X2Face does not require any further training for a new identity. However, for a fair comparison with vid2vid, we first optimised it on the multiperson database, on the task of PNCC-to-frame translation. After that, we fine-tuned it on five new identities, using 50 frames. We optimised our model with the same data. We evaluated the quality of synthetic samples on the task of video reconstruction (self-reenactment). We present our findings in Table 3.4. As can be seen, the average pixel distance between generated and ground truth is significantly smaller for our samples. In Fig. 3.10 we present a visual comparison with the aforementioned systems.

[1] https://github.com/zllrunning/face-parsing.PyTorch#Demo

Table 3.4: Numerical comparison of our proposed Head2Head extension with X2Face [WKZ18] and vid2vid [WLZ+18a] on the task of reconstruction, for five different target videos-identities. We employ the average pixel distance to measure the fidelity of reconstruction, which reveals that our method outperforms the two baselines.

Average Pixel Distance (↓)			
Test video-identity	X2Face [WKZ18]	vid2vid [WLZ+18a]	Ours
1	18.05	21.77	**14.62**
2	15.48	19.11	**13.09**
3	**13.42**	23.08	15.21
4	26.54	35.23	**25.74**
5	26.27	22.76	**20.34**



Figure 3.10: Our approach generates samples of higher quality samples when compared to X2Face [WKZ18] and vid2vid [WLZ+18a].

Ablation study First, we examine the significance of the initial training stage using a multiperson database. To that end, we conducted an ablation study, using four training videos for fine-tuning, with the identities illustrated in Fig. 3.11. We tested a variation of our proposed method without the initial learning phase. That is, we initialised the parameters of all networks before fine-tuning with random values. For the first two identities in Fig. 3.11 we utilised target sequences of 500 frames each. For the other two identities we used 250 frames instead. In Table 3.5 we report the average pixel distance between synthetic and ground truth test frames. The numbers reveal that the initialisation stage is very crucial. This is confirmed visually in Fig. 3.11, as the model variation without initialisation generates non-realistic samples with severe artifacts.

Table 3.5: Average pixel distance between generated and ground truth samples. We experimented with four videos-identities and tested our model with or without the initial learning stage. The reported numbers show the significance of the first training stage.

Average Pixel Distance (↓)		
Test video-identity	Ours	Ours w/o initialisation
1	**29.44**	36.61
2	**22.21**	34.52
3	**18.72**	34.92
4	**14.40**	29.19

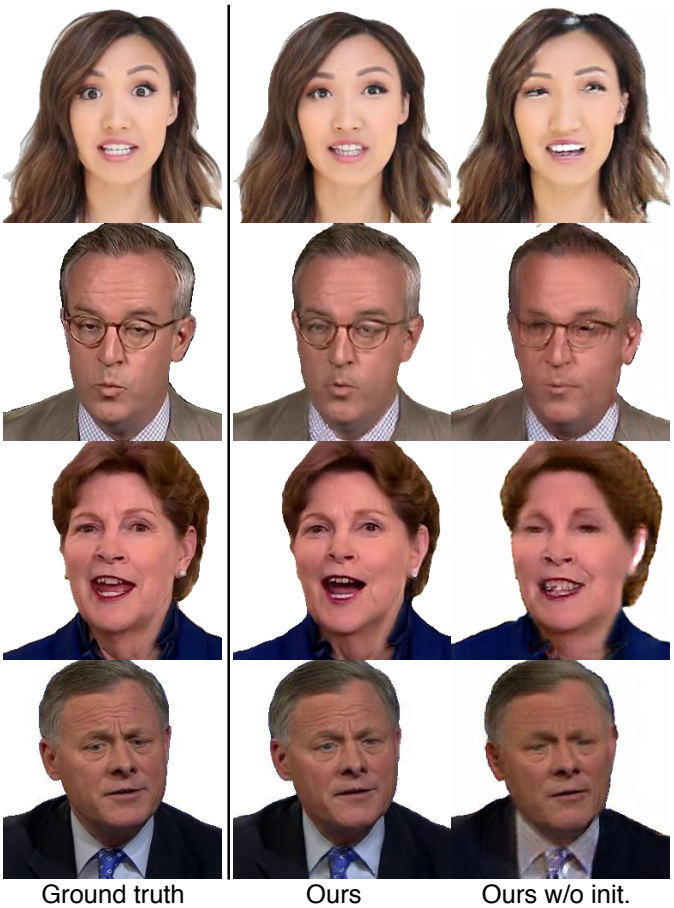


Figure 3.11: Visual comparison of our generated samples with and without the initial learning step.

Following up, we measure the contribution of the dynamics and mouth discriminator, D_V and D_M, to the performance of network G. For that reason, we prepared three variations of our model: without D_M, without D_V, without D_M and D_V. Afterwards, we calculated the average pixel distance for six test videos-identities during self-reenactment. We report the results in Table 3.6, which reveal that removing the dynamics discriminator reduces the quality of samples by a wide margin. Moreover, the variation without a mouth discriminator also produces lower quality frames, while removing both networks has an even larger impact on the average pixel distance.

Test video-identity	**Average Pixel Distance (↓)**			
	Ours	w/o D^M	w/o D^V	w/o D^V, D^M
1	**28.51**	30.01	39.93	41.80
2	**17.86**	19.93	38.97	25.73
3	**25.96**	29.71	49.42	57.27
4	**27.90**	49.22	48.73	53.57
5	**25.11**	34.61	41.48	43.31
6	**29.05**	31.30	39.34	45.96

Table 3.6: Quantitative results for the significance of networks D^M and D^V.

Last but not least, we measure the accuracy of the generative network when it comes to the prediction of the foreground mask. For that, we conducted the following experiment: We performed a self-reenactment with five videos-identities and then calculated the mean Intersection over Union (IoU) between the predicted and ground masks. In Fig. 3.12 we present the IoU for each video. As can be seen, the foreground masks estimated by network G are very accurate.

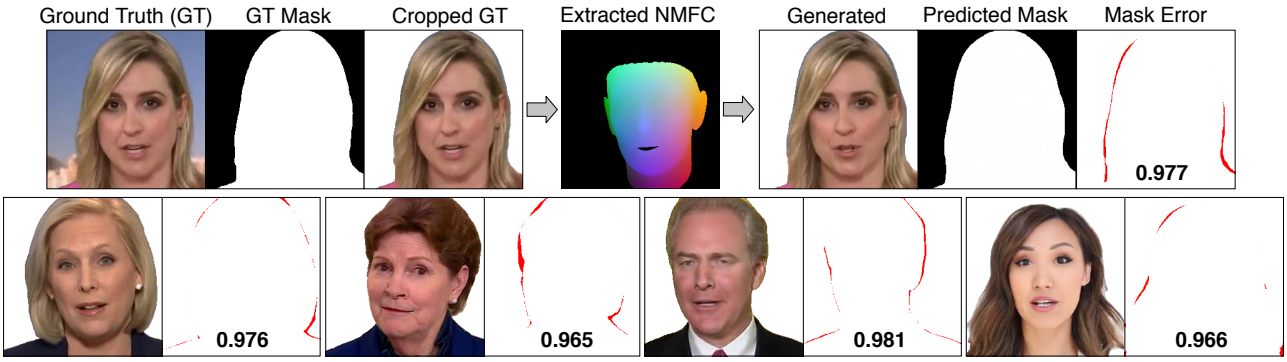


Figure 3.12: Mean Intersection over Union (IoU) for five different videos-identities.

3.5 Summary

Generating photo-realistic video portraits of human faces in a controllable manner is a challenging task. Recently, full head reenactment methods have shown incredible results when it comes to transferring the facial expressions, head movements and the eye gaze from a source actor to another target identity. In this chapter, we presented Head2Head, a video-based approach to the problem that utilises a 3D face reconstruction pipeline and relies on concepts from deep generative modeling and GANs. More specifically, we adopt a 3D parametric representation of

the face along with a 3DMM fitting algorithm for video portraits. The use of 3DMMs helps to disentangle expression from identity-related characteristics. First, we perform 3D face reconstruction and map a source input video to a sequence of PNCC images, a representation that holds semantic facial information. After that, a video rendering neural network translates the PNCC images into photo-realistic frames of a target individual, by completing the facial texture, as well as the hair, torso and background areas. To that end, we employ an person-specific GAN architecture from video-to-video translation and pay attention to the mouth area, using a dedicated mouth discriminator network. Our proposed method generates frame sequences that are temporally stable, as it relies on FlowNet 2.0 [IMS+17] fine-tuned for facial optical flow estimation. We urge the reader to refer to our supplementary video, since our synthetic data are better viewed as videos. Last but not least, we extended our method to accommodate synthesis of multiple identities, with the assistance of a few shot learning strategy and a novel architecture, based on adaptive normalisation layers.

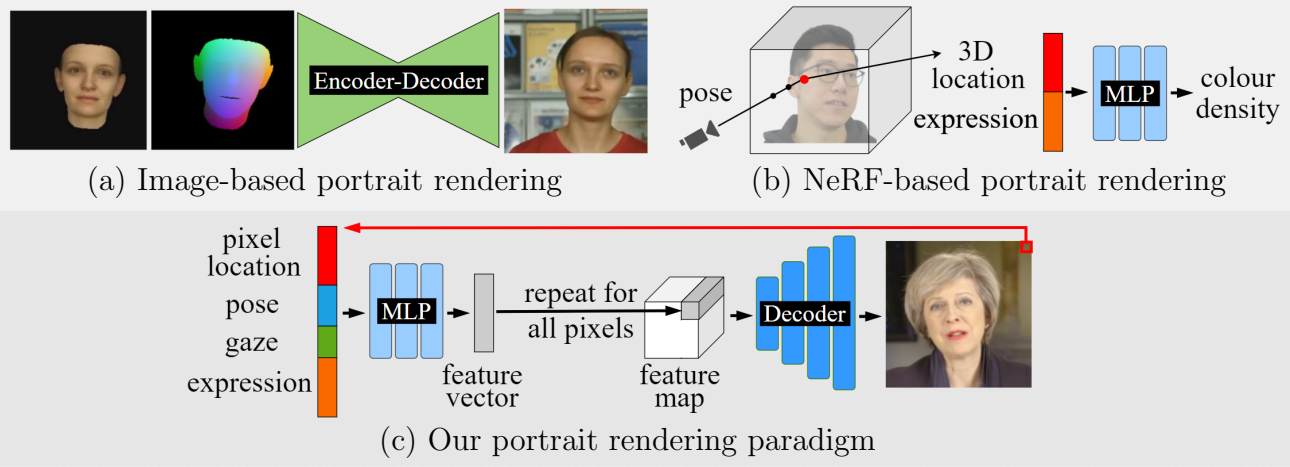
Chapter 4

Dynamic Neural Portraits

Recent attempts to controllable video portraits synthesis using neural networks have shown very promising results. Several learning-based approaches to video portrait generation rely on Generative Adversarial Networks (GANs) [GPAM+14] and image-to-image translation models [IZZE17]. For instance, Deep Video Portraits (DVP) [KGT+18] utilise a neural network that maps coloured renderings of 3D faces into face portraits. Differently, our method presented in Chapter 3, Head2Head [KDRZ20], follows a video-based approach for translating Projected Normalized Coordinate Code (PNCC) [ZLL+16] renderings into photo-realistic images. Both of the aforementioned face renderings depend on expression blendshapes, retrieved by fitting 3D Morphable Models (3DMMs) [BV99, PKA+09, CWZ+13, BRP+18] to video data. The fitting step is followed by a rendering step, which prepares the image renderings of 3D faces.

More recently, NerFACE [GTZN21] capitalised on the photo-realism achieved by Neural Radiance Fields (NeRFs) [MST+20] and produced video portraits of unprecedented quality in larger resolutions, such as 512^2. Nonetheless, due to ray casting and volume sampling, image rendering with NerFACE is notably slow, requiring several seconds per frame. Moreover, the authors make the assumption that the tracked head pose parameters coincide with the camera viewpoints of the scene, which causes significant inconsistencies in torso synthesis, as in practice the camera is stationary and therefore only a single viewpoint of the scene actually exists. Although their modeling provides multi-view consistency for the face region, it violates consistency for the upper torso and results in severe trembling. AD-NeRF [GCL+21] proposes an audio-driven method for portrait synthesis and solves this problem by taking advantage of body part segmentation maps [LLWL20]. They split the portrait into head, torso and background and devise two separate NeRF models, one for the head that uses head poses as camera viewpoints, and a second one for the torso that treats head poses as simple inputs to the MLP, while camera viewpoints are considered fixed. However, AD-NeRF requires many seconds to render a single frame in 512^2, since during inference it evaluates two MLPs instead of one.

In this chapter, we take a completely different approach from previous full head reenactment methods and present **Dynamic Neural Portraits**, a fast and efficient method for controlling video portraits. Our system is inspired by implicit neural representations (INR), as we leverage neural networks to parameterise video portraits. We follow practices from conditionally independent pixel synthesis (CIPS) [ADK+21] and 2D image-based neural rendering with convolutional networks. Our model is built on top of a 2D coordinate-based MLP with controllable dynamics, as we condition an MLP network on temporal information, such as expression,



(a) Image-based portrait rendering (b) NeRF-based portrait rendering

(c) Our portrait rendering paradigm

Figure 4.1: In contrast to traditional image-to-image translation approaches for portrait rendering, such as DVP [KGT+18] and Head2Head [KDRZ20] (a), or recent NeRF-based portrait rendering methods, namely NerFACE [GTZN21] and AD-NeRF [GCL+21] (b), we propose Dynamic Neural Portraits, a novel architecture paradigm for video portrait synthesis (c).

pose and gaze parameters, without a need for renderings of 3D faces as inputs. To the best of our knowledge, we are the first to condition a 2D coordinate-based MLP on expression, pose and gaze parameters to explicitly control face synthesis. Nonetheless, instead of directly predicting colours, our MLP network outputs feature vectors, which are accumulated for all 2D spatial locations and up-sampled with a CNN-based decoder, in a similar way to 3D aware GANs [NG21, GLWT21, ZXNT21]. We optimise the MLP and decoder networks jointly, in an end-to-end fashion. Moreover, unlike previous methods that focus exclusively either on expression blendshapes or audio signals as driving data, our work demonstrates how to leverage both modalities using the same architecture. Our approach combines high-quality results with unparalleled execution speed, 270 times faster than related NeRF-based state-of-the-art methods for reenactment [GTZN21, GCL+21]. The contributions presented in this chapter can be summarised as follows:

- We propose a generative model that consists of a 2D coordinate-based MLP with controllable dynamics and a CNN-based decoder, as a new approach to reenactment. Our method does not rely on GANs or NeRFs.
- We show that this novel architecture can be driven either by expression blendshapes or audio-based features, with small modifications.
- Our comprehensive experiments demonstrate the merits of our method, as it outperforms recent state-of-the-art systems, both in terms of execution speed and photo-realism.

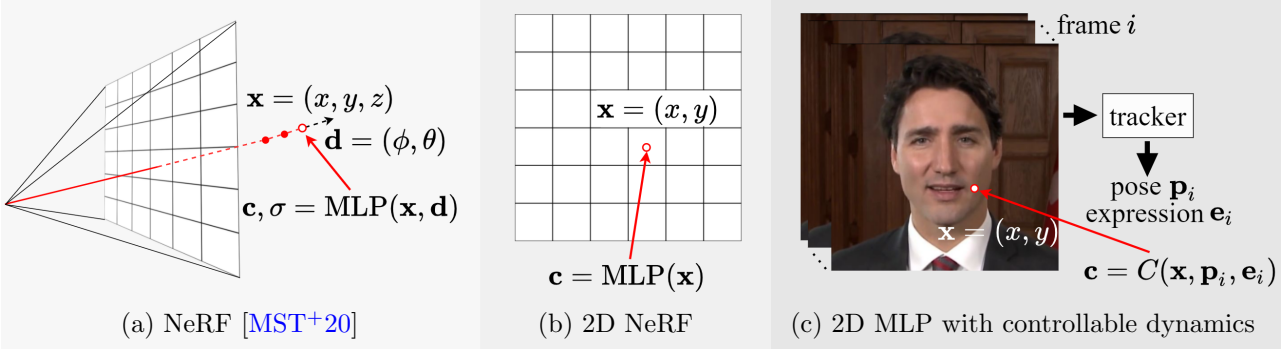
In Section 4.1 we describe our baseline reenactment approach, a 2D coordinate-based MLP with controllable dynamics. Next, in Section 4.2, we build upon this baseline and present our proposed method, namely Dynamic Neural Portraits. In Section 4.2.1 we show an extension of our system that supports audio-driven portrait synthesis and in Section 4.2.2 we discuss the optimisation objective. Then, in Section 4.3, we present the experimental evaluation of our system and demonstrate the improvements it achieves over prior state-of-the-art methods, for the tasks of self-reenactment, cross-identity reenactment and audio-driven portrait synthesis.

4.1 2D-coordinate MLP with Controllable Dynamics

NeRF [MST+20] represents a static scene as a continuous function of 3D location coordinates $\mathbf{x} = (x, y, z)$ and the camera ray's viewing direction $\mathbf{d} = (\theta, \phi)$. This function outputs the emitted radiance (colour) $\mathbf{c}$ as well as the density σ at this location. In practice, the underlying volumetric scene function is parameterised by an MLP. The scene can be rendered under different camera viewpoints using principles from classical volume rendering [KVH84, Lev90, Max95]. In order to render the scene from a given viewpoint, first we cast a ray and evaluate the MLP in 3D locations crossed by the ray. Then, we accumulate the colours and densities into an image, with hierarchical volume sampling [MST+20]. NeRF is optimised using images of the static scene and the corresponding intrinsic and extrinsic camera parameters, by minimising the error between the rendered and true pixel colours.

A simple reduction of NeRF can be done by replacing the 3D location input with 2D coordinates $\mathbf{x} = (x, y)$, thus converting the underlying volume into a plane. Now, since a 2D space has taken the place of the 3D volume, the viewing direction input of the original NeRF function is no longer meaningful and thus can be removed. In this way, we end up with a 2D vector-valued function that instead of static scenes is limited to represent static 2D data, such as image portraits. This continuous portrait representation can be modeled with an MLP network, which learns to predict the pixel colour $\mathbf{c}$ at a given input location $\mathbf{x}$. 2D NeRF can be seen as a simplification of NeRF, when a single viewpoint of a scene is available, as the camera remains fixed. Instead of relying on slow hierarchical volume sampling, a 2D coordinate-based MLP has the advantage of rendering images much faster, simply by evaluating the MLP at each spatial location of the image plain.

The 2D coordinate-based MLP described above is limited to reconstructing a single image from its pixel coordinates. Following an approach similar to dynamic NeRF models [GTZN21, PCPMMN21] for representing deformable scenes, we extend the 2D coordinate-based MLP for handling temporal data, such as video portraits of faces. Here, we focus on videos captured by a monocular stationary camera. A time-varying representation can be obtained by conditioning the MLP network on information that varies between frames. Given a sequence of frames $\mathbf{I}_{1:T}$ and the corresponding head pose $\mathbf{p}_{1:T}$ and facial expression parameters $\mathbf{e}_{1:T}$, which have been regressed by a face tracking system, an intuitive solution would be to estimate the pixel's RGB



(a) NeRF [MST+20] (b) 2D NeRF (c) 2D MLP with controllable dynamics

Figure 4.2: Our baseline model, namely 2D-coordinate MLP with controllable dynamics (c), is based on 2D NeRF (b), a simplification to NeRF (a) after replacing 3D spatial coordinates with 2D.

value at position $\mathbf{x}$ of the i-th video frame with an MLP network, as

$$\mathbf{c} = C(\mathbf{x}, \mathbf{p}_i, \mathbf{e}_i). \tag{4.1}$$

Here, $\mathbf{c}$ is the colour at 2D location $\mathbf{x} \in [0,1]^2$, $\mathbf{p}_i \in \mathbb{R}^6$ is the head pose given by rotation (Euler angles) and translation parameters and $\mathbf{e}_i \in \mathbb{R}^{n_{exp}}$ are facial expression blendshape parameters. Please note that the pose parameters $\mathbf{p}_i$ differ from the extrinsic camera parameters in static scenes represented by NeRF-like models, as they describe the pose of the face (without the neck and torso), while the camera position is considered fixed. Apart from modeling head movements and expression changes in video frames during training, this parameterisation enables to explicitly control the 2D coordinate-based MLP, using external parameters $\mathbf{p}_{ext}, \mathbf{e}_{ext}$ of our choice. That is, during inference we may reenact our portrait using pose and expression information extracted from a source video of a different identity.

4.2 Dynamic Neural Portraits

Despite the fact that a 2D coordinate-based MLP with controlable dynamics is a straightforward approach for modeling video portraits with a single MLP network, in practice we found that the photo-realism of generated samples and the rendering speed significantly degrade as we increase the video resolution. As we show in the experimental section, we obtain superior results in terms of visual quality by combing the MLP with a convolutional decoder network. Following the paradigm of recent 3D aware GANs [NG21, GLWT21, ZXNT21], instead of predicting RGB colour values we propose an MLP that maps its input to a visual feature vector $\mathbf{f} \in \mathbb{R}^{n_f}$. Moreover, we acquire more accurate eye movements by adding gaze angles $\mathbf{g}_i \in \mathbb{R}^2$ as an extra input to our MLP network. Given the 2D spatial location $\mathbf{x}$ along with pose $\mathbf{p}_i$, gaze $\mathbf{g}_i$ and expression information $\mathbf{e}_i$, our MLP now predicts a vector of features

$$\mathbf{f} = F(\gamma(\mathbf{x}), \gamma(\mathbf{p}_i), \gamma(\mathbf{g}_i), \mathbf{e}_i, \mathbf{v}_i). \tag{4.2}$$

We noticed that introducing *per-image learnable latent variables* $\mathbf{v}_i$ as inputs, a technique which has been previously adopted in NeRF methods [MBRS+21, PSB+21, PSH+21, GTZN21], improves the stability of our network, since it enables the MLP to learn variations among frames that are not modeled by the pose and expression variables (e.g. torso movements, illumination changes, small background motions). Furthermore, we follow established practices of previous NeRF-based systems and adopt positional encoding to the MLP inputs, and more specifically on the position $\mathbf{x}$, pose $\mathbf{p}_i$ and gaze $\mathbf{g}_i$ vectors. The standard encoding function

$$\gamma(x) = [x, \sin(2\pi x), \cos(2\pi x), \ldots, \sin(2^{N-1}\pi x), \cos(2^{N-1}\pi x)]^\top \tag{4.3}$$

originally proposed by [MST+20] is applied to all values of $\mathbf{x}$, $\mathbf{p}_i$ and $\mathbf{g}_i$, mapping each number $x \in \mathbb{R}$ to a higher dimension embedding $\gamma(x) \in \mathbb{R}^{2 \cdot N+1}$. Replacing the original inputs with their embeddings allows our model to achieve high frequency details in the generated images.

The network described above predicts a feature vector for a given 2D spatial location, conditioned on head pose, facial expression and eye gaze information. In order to render frame i, first we evaluate our MLP network at all spatial locations $\mathbf{x} \in \mathbf{X}$ that correspond to resolu-

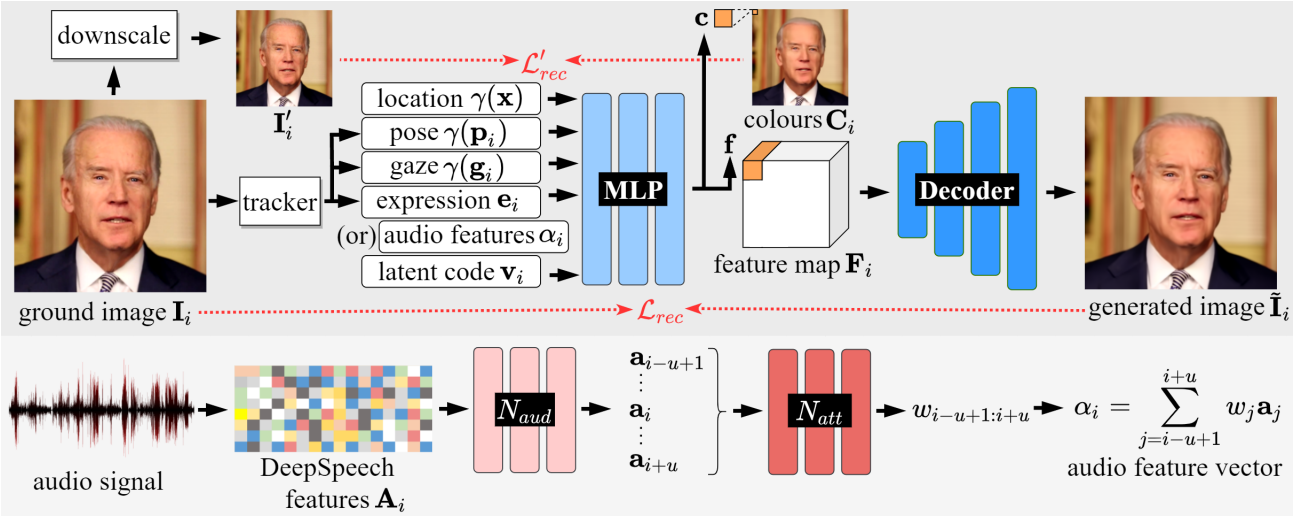


Figure 4.3: An overview of Dynamic Neural Portraits method.

tion $H_f \times W_f$ and accumulate the resulting features in a visual feature map $\mathbf{F}_i \in \mathbb{R}^{H_f \times W_f \times n_f}$. Then, a decoding network D receives the feature map and performs up-sampling, in order to synthesise the output frame $\tilde{\mathbf{I}}_i = D(\mathbf{F}_i)$ at the target resolution $H \times W$, as shown in Fig. 4.3.

4.2.1 Audio-driven Portrait Synthesis

Our choice to inject the driving signals (i.e. pose, expression and gaze parameters) into our system through an MLP network enables to easily adapt our method to other modalities, such as audio. Unlike previous works [GCL+21, GTZN21, TET+20, KGT+18], which focus exclusively either on expression blendshapes or audio data as driving signals, we show that the same architecture can be used effectively for both modalities. More specifically, we can replace the expression blendshapes with an audio feature vector $\boldsymbol{\alpha}_i$ as input to the MLP network, in case the audio stream is available. We extract high-level features from acoustic signals with the widely-adopted DeepSpeech model [HCC+14, AAA+16]. As a first step, we assign a 29-dimensional vector estimated by DeepSpeech to every video frame. Then we create a window of vectors with size $w = 16$ around each frame, taken from its neighboring (past and future) time steps. In this way each frame i is coupled with a DeepSpeech feature $\mathbf{A}_i \in \mathbb{R}^{16 \times 27}$. We follow a similar approach with AD-NeRF [GCL+21] and utilise a 1D-convolutional network N_{aud} that learns to compute per-frame latent codes $\mathbf{a}_i \in \mathbb{R}^{n_a}$ from $\mathbf{A}_i$. Next, we employ a self-attention network N_{att}, as proposed in [TET+20] and [GCL+21], which operates as a temporal filter on subsequent audio codes $\mathbf{a}_{i-u+1:i+u}$ and mixes them up with the assistance of predicted attention weights $w_{i-u+1:i+u}$, to form the final audio feature vector $\boldsymbol{\alpha}_i = \sum_{j=i-u+1}^{i+u} w_j \mathbf{a}_j$. We set $u = 4$, which results in a window of $2u = 8$ time steps.

4.2.2 Objective Function and Optimisation

We train our proposed model on the task of image reconstruction. Given the generated image $\tilde{\mathbf{I}}_i$ and the corresponding ground truth frame $\mathbf{I}_i$, the reconstruction loss is equal to the total squared error between the predicted and ground pixel colour, for each pixel. We observed that

we can further improve our method's reconstructive performance by adding an extra output layer to the MLP, which predicts the colour $\mathbf{c} \in \mathbb{R}^3$ side-by-side with visual features $\mathbf{f}$, and minimising the distance of the predicted colour from the true one. To that end, we accumulate the colour outputs for all 2D spatial locations in $\mathbf{C}_i \in \mathbb{R}^{H_f \times W_f \times 3}$, and penalise its distance from the ground truth image. Please note that in practice we down-scale the true image to resolution $H_f \times W_f$, in order to match the resolution of $\mathbf{C}_i$, which yields $\mathbf{I}'_i$. The overall loss function for frame i is given by

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}'_{rec} = ||\tilde{\mathbf{I}}_i - \mathbf{I}_i||_2^2 + ||\mathbf{C}_i - \mathbf{I}'_i||_2^2 \tag{4.4}$$

The MLP network F and the CNN-based decoder network D are optimised jointly. In case we drive synthesis with audio, networks N_{aud} and N_{att} are also included in the optimisation process. We optimise a separate model for each video portrait.

4.3 Experiments

4.3.1 Implementation Details

Face tracking Our proposed method is primarily driven using head pose, facial expression and eye gaze information. We employ [KRS+19] to extract gaze angles from each frame of a given video portrait. We recover expression blendshapes using two different approaches, one for reconstruction (self-reenactment) and another one for cross-identity motion transfer (reenactment) experiments. We observed that GANFIT [GPKZ19] operates very well when it comes to expression and shape (identity) disentanglement, thus we use this tracker throughout our cross-identity reenactment experiments. Nonetheless, during self-reenactment, we use the approach presented in [DZS21], which relies on dense 3D facial landmarks regressed by RetinaFace [DGV+20a] and 3DMM fitting on LSFM morphable model [BRP+18], as it appears to retain more detailed expression information from RGB frames in comparison to GANFIT. Moreover, we employ the same tracking method with [GCL+21] to retrieve head pose parameters.

Networks Architecture Our proposed MLP network consists of eight consecutive linear layers, each followed by a ReLU activation function. Similarly to NeRF [MST+20], we use a skip connection in the fourth layer. The MLP's input vector emerges from the concatenation of

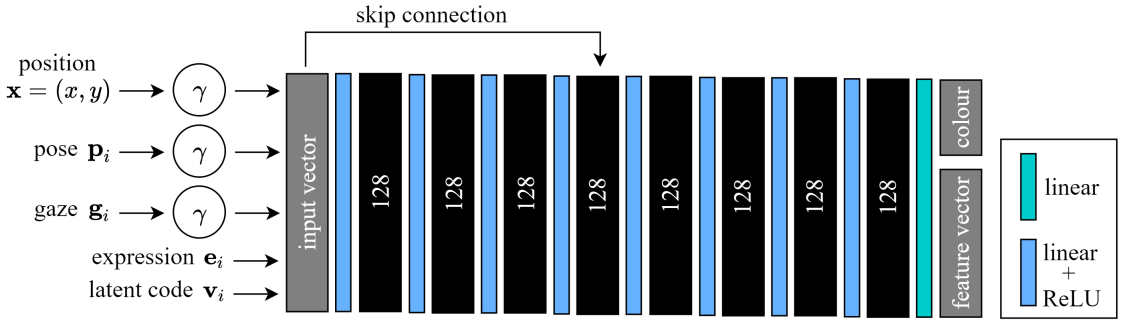


Figure 4.4: The architecture of our MLP network is based on the one originally proposed for NeRF [MST+20], with the addition of a linear output layer that returns the RGB colour $\mathbf{c}$ and feature vector $\mathbf{f}$. Here, γ denotes the function of positional encoding.

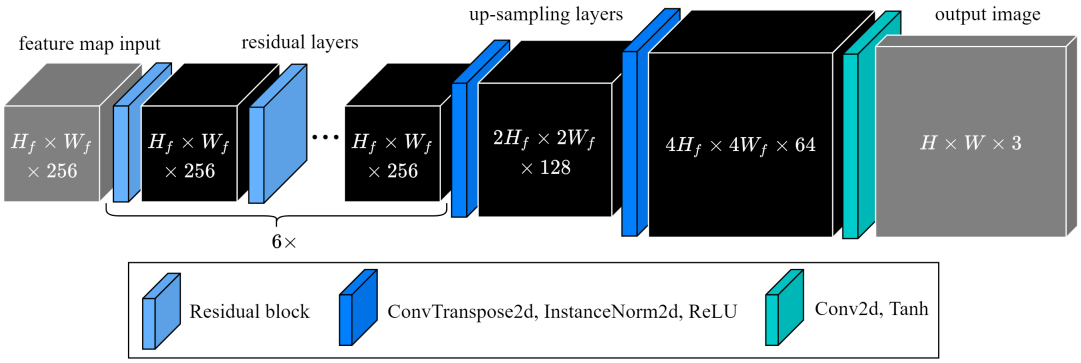


Figure 4.5: The architecture of our decoder is based on the generator of pix2pixHD [WLZ+18b], with six residual layers followed by two up-sampling layers and a final output layer. For more details on the structure of the residual block, please refer to [WLZ+18b].

the position $\gamma(\mathbf{x})$, head pose $\gamma(\mathbf{p}_i)$ and gaze $\gamma(\mathbf{g}_i)$ parameters, all of which are position encoded, as well as the expression parameters $\mathbf{e}_i$ and learnable latent code $\mathbf{v}_i$. In the output, a simple linear layer predicts colour $\mathbf{c}$ and feature vector $\mathbf{f}$. Please see Fig. 4.4 for more details. Our adopted image decoder is based on the decoding part of the generator proposed originally in pix2pixHD [WLZ+18b]. It receives a feature map of resolution $H_f \times W_f$, assembled from the feature vectors created by the MLP, and uses it to synthesise a photo-realistic image. It is made up of six consecutive residual layers followed by two up-sampling layers and a final output layer that hallucinates the output image of resolution $H \times W$, where $H = 4H_f$ and $W = 4W_f$. The architecture of our CNN-based decoding network is displayed in Fig. 4.5. For the architecture of audio-related networks, please refer to [GCL+21].

Dataset and Training Our method is optimised on monocular RGB videos, of various resolutions: 256×256, 512×512 and 1024×1024. As a pre-processing step, frames are cropped around the target face. We experiment with sequences of lengths from 5K up to 20K frames. We maintain the original train and test split for all video data we acquired for comparison with other methods. As can be seen in Table 4.1, we used eleven different video portraits throughout use qualitative and quantitative experiments. We denote the number of frames we used to train each person-specific model by N_{train}, while N_{test} represents the number of test frames. Training frames come for the beginning of videos and test frames from the end, as we maintain the original training and test split of data.

For training, we utilise ADAM [KB14] with $\beta_1 = 0.9, \beta_2 = 0.999$ and learning rate $\eta = 0.0005$. We train a new model for each portrait, which requires 100K to 150K iterations (approximately 18-27 hours) on a single NVIDIA Tesla V100 PCIe 32 GB, depending on the number of training frames.

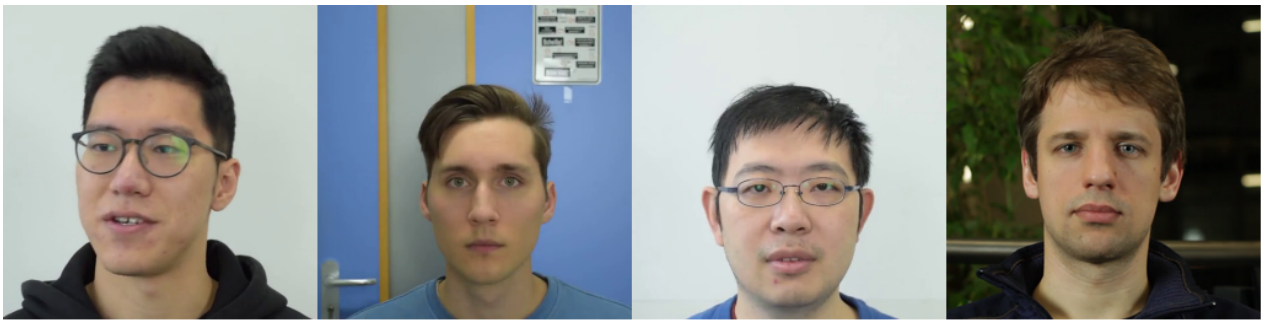
4.3.2 Evaluation Metrics

Here, we provide details on the evaluation metrics that we used in order to validate of our model's performance and compare it with state-of-the-art methods.

L1-distance This is the most basic means of evaluating the reconstructive performance of methods. Given a ground frame $\mathbf{I}_i \in [0, 255]^{H \times W \times 3}$ and the corresponding generated frame $\tilde{\mathbf{I}}_i$, L1-distance is given by $||\mathbf{I}_i - \tilde{\mathbf{I}}_i||_1$.

Table 4.1: Details of video portraits used throughout the experiments

Portrait	Resolution	N_{train}	N_{test}	Experiments	Data Source
ID. 1	512 × 512	5513	1000	reconstruction	NerFACE [GTZN21]
ID. 2	512 × 512	5518	1000	reconstruction	NerFACE [GTZN21]
ID. 3	512 × 512	5441	1000	reconstruction	NerFACE [GTZN21]
ID. 4	256 × 256	0	2380	reenactment	DVP [KGT+18]
ID. 5	256 × 256	18138	0	reenactment	DVP [KGT+18]
Obama	512 × 512	7272	728	audio-based reconstruction ablation study	AD-NeRF [GCL+21]
May	512 × 512	5500	550	audio-based reconstruction	Head2Head [KDRZ20]
Biden	512 × 512	5301	454	ablation study supplementary video	Head2Head [KDRZ20]
Trudeau	512 × 512	0	575	supplementary video	Head2Head [KDRZ20]
Macron	1024 × 1024	0	1374	supplementary video	Head2Head [KDRZ20]
Mitsotakis	1024 × 1024	10934	0	supplementary video	Head2Head [KDRZ20]



(a) ID. 1 (b) ID. 2 (c) ID. 3 (d) ID. 4



(e) ID. 5 (f) Obama (g) May (h) Biden



(i) Macron (j) Mitsotakis (k) Trudeau

Figure 4.6: Illustration of the different identities (videos) we used throughout our experiments.

LPIPS By its definition, L1 is a simple shallow function that assesses reconstruction only at pixel level. It has been shown that using feature maps for the evaluation of image reconstruction is more consistent with human perception. These feature maps are extracted from images with the assistance of pre-trained deep neural networks. Here, we employ the widely-used LPIPS [ZIE+18] metric as a perceptual similarity score between the ground $\mathbf{I}_i$ and synthetic frames $\tilde{\mathbf{I}}_i$, which is calculated based on visual feature maps computed with AlexNet [KSH12].

FID We use FID score [HRU+17, Sei20] in order to measure the similarity between the dataset of ground truth frames and the dataset of frames generated by a model. This score provides an insight into the visual quality and photo-realism of synthesised frames, as it is an established means of measuring the quality of images created by generative models, such as GANs [GPAM+14].

FVD Since we actually generate video data, it is essential to evaluate the performance of different methods based on a metric that takes into account the temporal coherence among frames. For that, we assess the plausibility of generated videos with FVD [UvSK+18], as this metric has shown to correlate well with human judgment on visual quality and photo-realism of video data.

CSIM This is a metric for calculating identity preservation during reenactment. Given that cross-identity motion transfer entails the problem of transferring characteristics of the driving identity to the generated frames of the target person, it is of upmost importance to calculate how well different methods handle identity preservation. For that, we use an identity recognition network, namely ArcFace [DGXZ19], for the computation of embedding vectors from the generated and real images of the target subject. Then, we compute the cosine similarity between embedding vectors coming from real and synthetic frames.

Expression Distance One of the most important aspects of reenactment lies in successful expression transfer. In order to evaluate the generated expressions numerically, we employ DECA [FFBB21] for the extraction of expression parameters from driving and generated frames. Then, we compute the L1-distance between each pair of expression vectors, coming from the driving and synthetic data.

Pose Distance Similarly to expression, we validate the accuracy of systems on head pose transfer. Again, we pass the driving and corresponding generated frames through DECA [FFBB21]. This yields a sequence of driving head rotation matrices $\mathbf{R}_t$ and synthetic head rotation matrices $\tilde{\mathbf{R}}_t$, $t = 1, \ldots, T$. Then, we compute the head rotation distance for frame t as

$$\theta_{pose} = \arccos\Big(\frac{\mathrm{tr}(\tilde{\mathbf{R}}_t \mathbf{R}_t^\top) - 1}{2}\Big) \tag{4.5}$$

and convert angle θ_{pose} to degrees.

Gaze Distance As suggested by the uncanny valley effect, accurate gaze synthesis is very important when creating realistic human faces. We measure gaze transfer on the task of reenactment by extracting the driving gaze vectors $\mathbf{g}_t$ and synthetic gaze vectors $\tilde{\mathbf{g}}_t$, $t = 1, \ldots, T$, with the assistance of a state-of-the-art gaze detection system [KRS+19]. After that, we measure the distance between gaze vectors as the angle between them:

$$\theta_{gaze} = \arccos\Big(\frac{\mathbf{g}_t^\top \tilde{\mathbf{g}}_t}{||\mathbf{g}_t||_2 ||\tilde{\mathbf{g}}_t||_2}\Big). \tag{4.6}$$

Finally, we convert θ_{gaze} to degrees.

4.3.3 Comparison with State-of-the-Art

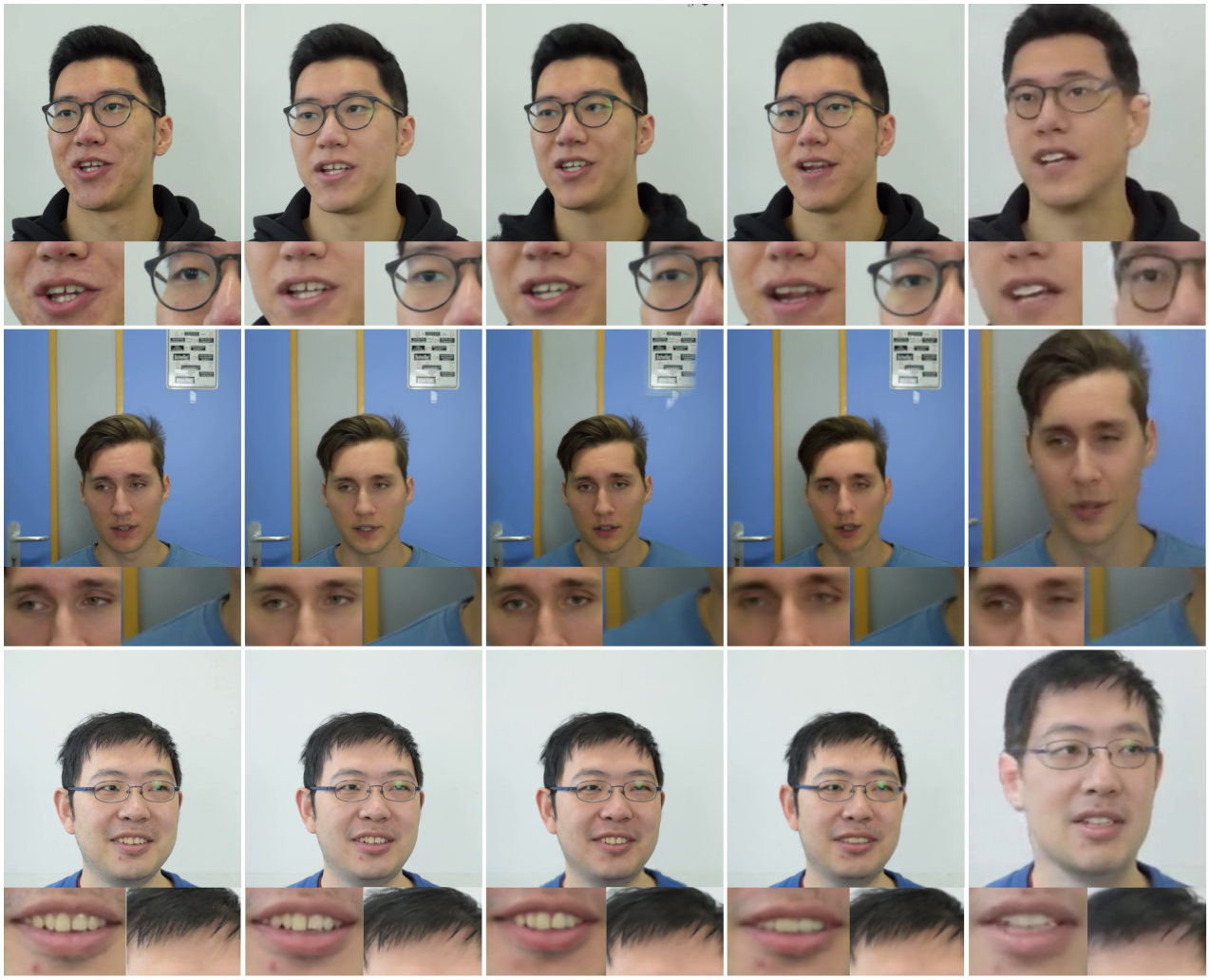
Reconstruction First, we present a numerical comparison with FOMM [SLT+19b], as a representative of person-generic models, as well as DVP [KGT+18] and NerFACE [GTZN21], the best-performing person-specific reenactment methods. We evaluate the generative performance of methods on the task of reconstruction, also know as self-reenactment. The results displayed in Table 4.2 indicate that our method outperforms all baselines, in the three different video portraits. Please note that the reported scores refer to the average metric values, computed across all test frames for each video portrait. An exception to that are FID [HRU+17] and FVD [UvSK+18] scores, as they are not computed on image pairs, but pairs of videos.

Table 4.2: Numerical comparison with FOMM [SLT+19b], DVP [KGT+18] and NerFACE [GTZN21] for three different portraits on the task of reconstruction (self-reenactment).

Method	Resolution	Portrait	L1 (↓)	LPIPS (↓)	FID (↓)	FVD (↓)
FOMM [SLT+19b]	256 × 256	ID. 1	8.79	0.095	25.48	331.0
		ID. 2	7.60	0.084	37.24	338.3
		ID. 3	9.48	0.130	24.72	254.10
DVP [KGT+18]	256 × 256	ID. 1	6.95	0.152	49.35	195.4
		ID. 2	9.08	0.079	37.58	464.3
		ID. 3	8.01	0.123	51.30	196.7
NerFACE [GTZN21]	512 × 512	ID. 1	**6.19**	0.136	74.22	278.3
		ID. 2	10.98	0.143	74.02	357.7
		ID. 3	6.28	0.067	34.64	81.17
Ours	512 × 512	ID. 1	6.45	**0.094**	**23.76**	**169.8**
		ID. 2	**5.21**	**0.071**	**24.60**	**222.1**
		ID. 3	**5.15**	**0.051**	**23.94**	**78.06**

We assess the reconstructive performance of systems quantitatively, with the assistance of L1-distance between the generated and ground truth test frames, as well as Learned Perceptual Image Patch Similarity (LPIPS) [ZIE+18], which tests the perceptual similarity between images. Moreover, we determine photo-realism with Fréchet Inception Distance (FID) [HRU+17] and Fréchet Video Distance (FVD) [UvSK+18] metrics, which appear to correlate well with human perception. All scores suggest that our approach produces superior samples in terms of visual quality and photo-realism. This observation is confirmed visually in Fig. 4.7. As can be seen, our method generates more crispy images with finer details and more consistent gaze.

Reenactment As opposed to previous works [KGT+18, GTZN21], we further validate our model's performance on the task of cross-identity motion transfer (reenactment) in a quantitative way. This task involves passing on the head pose, facial expression and eye gaze from a driving video to the target actor, while preserving the identity characteristics, as the two subjects are now different. We compare our approach with DVP [KGT+18], a system which



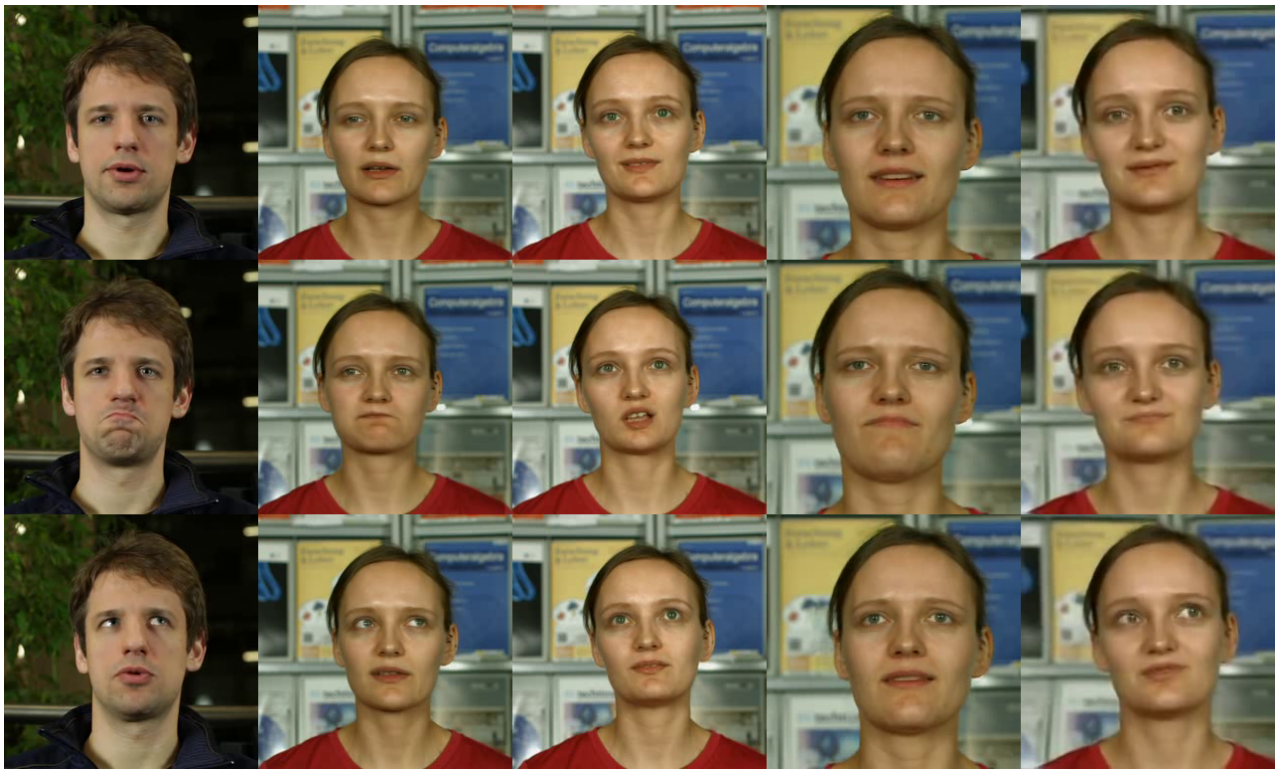
ground truth ours NerFACE [GTZN21] DVP [KGT+18] FOMM [SLT+19b]

Figure 4.7: Visual comparison with the baselines on the problem of reconstruction (self-reenactment). Our method consistently generates more realistic samples with finer details than its counterparts [GTZN21, KGT+18, SLT+19b]. Note that for the correct evaluation of FOMM, images needed to be further cropped, thus appearing zoomed in the face.

pays particular attention to shape (identity) and expression disentanglement. Moreover, we examine how well our algorithm operates against FOMM [SLT+19b] and HeadGAN [DZS21]. To that end, we use the source and target video portraits provided by the authors of DVP. In Table 4.3, we show that our model achieves better scores than all three baselines across all

Table 4.3: Numerical comparison with FOMM [SLT+19b], HeadGAN [DZS21] and DVP [KGT+18] on the task of cross-identity motion transfer (reenactment).

Method	CSIM (↑)	Expr. Dist. (↓)	Pose Dist. (↓)	Gaze Dist. (↓)
FOMM [SLT+19b]	0.840	13.28	6.31°	9.42°
HeadGAN [DZS21]	0.755	14.17	3.26°	6.35°
DVP [KGT+18]	0.861	11.95	4.93°	8.37°
Ours	**0.885**	**10.69**	**2.57°**	**4.74°**



driver ours DVP [KGT+18] HeadGAN [DZS21] FOMM [SLT+19b]

Figure 4.8: Visual comparison with baselines on reenactment. Our approach transfers pose, expression and gaze more reliably than both person-specific DVP [KGT+18] and person-generic methods HeadGAN [DZS21] and FOMM [SLT+19b]. Please note that for the correct evaluation of HeadGAN and FOMM, images had to be cropped closer to the face.

metrics related to successful reenactment. Additionally, in Fig. 4.8, we showcase examples of frames where our method appears to be more reliable than DVP [KGT+18], HeadGAN [DZS21] and FOMM [SLT+19b], in terms of pose, expression and gaze transfer. For this quantitative evaluation, we measure the target's identity preservation with Cosine Similarity (CSIM), based on identity embeddings extracted with the assistance of ArcFace [DGXZ19]. Moreover, we use DECA [FFBB21] to regress pose and expression parameters, both from the driving and generated frames. Then, we compute the L1-distance between expression parameters, as well as the head rotation distance in degrees. Finally, we use the gaze estimator from [CS18] to regress gaze vectors, and calculate their angular distance.

Audio-driven reconstruction Apart from conditioning our MLP network on expression blendshapes, we demonstrate the generative performance of our system when driven by acoustic signals. For that, we conducted a side-by-side comparison with AD-NeRF [GCL+21], the state-of-the-art model on this task.

Table 4.4: Numerical comparison with AD-NeRF [GCL+21] on audio-driven reconstruction.

Method	Portrait	L1 (↓)	LPIPS (↓)	FID (↓)	FVD (↓)
AD-NeRF [GCL+21]	Obama	4.11	0.083	16.67	225.7
	May	**5.13**	0.143	47.31	272.2
Ours	Obama	**4.04**	**0.054**	**9.12**	**110.3**
	May	5.72	**0.087**	**26.72**	**96.1**



ground truth ours AD-NeRF [GCL+21]

Figure 4.9: Visual comparison with AD-NeRF [GCL+21] on audio-driven reconstruction. We generate more accurate lips motions and higher quality details compared to AD-NeRF.

As shown in Table 4.4, our quantitative analysis reveals that our approach performs better than AD-NeRF, both in terms of reconstruction and image quality. Our findings can be observed also visually in Fig. 4.9.

Execution Speed Owing to its lightweight architecture, our method is able to render frames in a resolution up to 1024×1024, in 24 fps. For lower resolutions (e.g. 256×256 and 512×512) our pipeline operates in speeds faster than real time. In comparison to recent NeRF-based state-of-the-art methods [GTZN21, GCL+21], our system achieves a significant speed-up, generating images nearly 270 times faster.

Table 4.5: Comparison of the execution time of our generative model with various related methods. Time is reported in milliseconds (msec). We note that the results shown below refer to the forward pass time of models during inference, without considering data pre-processing such as face cropping, 3DMM fitting or audio features extraction.

Method	256×256		512×512		1024×1024	
	time (↓)	fps (↑)	time (↓)	fps (↑)	time (↓)	fps (↑)
AD-NeRF [GCL+21]	-	-	9630	0.10	-	-
NerFACE [GTZN21]	-	-	8465	0.12	-	-
DVP* [KGT+18]	65	15.4	196	5.1	-	-
HeadGAN [DZS21]	41	24.5	-	-	-	-
FOMM [SLT+19b]	21	47.2	-	-	-	-
Ours	**11**	**90.9**	**31**	**32.3**	**42**	**24.2**



ground truth our baseline (A) our full model (F)

Figure 4.10: Qualitative results of our ablation study. Our full model (F) (Dynamic Neural Portraits) improves the frames synthesised by our baseline (A) (2D coordinate-based MLP with controllable dynamics) by a noteworthy margin. This is clearly shown in the facial region of video portraits (eyes, teeth, etc.).

In Table 4.5, we report the execution times of baselines in different resolutions, measured on NVIDIA's Tesla V100 PCIe 32 GB. Please note that for DVP [KGT+18] we use the numbers presented in [KGT+18], measured on NVIDIA's GeForce GTX Titan Xp 12GB, as the source code of DVP is not publicly available*.

4.3.4 Ablation study

In this section, we evaluate our design choices and validate the significance of different components that make up our system. We conduct quantitative and qualitative experiments for six variations of our system. To that end, we test all variations on the task of reconstruction, for two separate portraits (Biden and Obama). We start with a 2D coordinate-based MLP with controllable dynamics, described in Section 4.1, which is used as the baseline model (A) of our ablation study. In order to demonstrate the advantages of our proposed architecture, we further experiment with a CNN-based decoder architecture only, without the MLP network (B). Then, we couple the MLP with the decoding network, which leads to variation (C). We form the next variations by first adding the learnable latent variables input (D), then including the reconstruction loss term $\mathcal{L}'_{rec}$ (E) and finally the gaze input (F), ending up with our full model as presented in Section 4.2. Our numerical analysis presented in Table 4.6 reveals the importance of each component, especially the huge impact of the MLP network when coupled with a CNN-based decoding network. As can be seen, the latent variables increase FVD scores as they help to stabilise movements between frames, while $\mathcal{L}'_{rec}$ improves the perceptual similarity with ground truth data, according to LPIPS. Finally, gaze input corrects eye motions, something that becomes more apparent when inspecting our generated video footage. In Fig. 4.10, we illustrate some examples of the visual differences between samples generated by the baseline model (A) and the full model (F) that we propose. Please note that for all the experiments we use positional encoding for the location and pose inputs of the MLP network.

Table 4.6: Quantitative results of our ablation study on two separate video portraits (Biden and Obama), of resolution 512×512.

Variation		Portrait	L1 (↓)	LPIPS (↓)	FID (↓)	FVD (↓)	Gaze Dist. (↓)
(A):	2D MLP contr. dyn. (baseline, Sec. 4.1)	Biden	5.33	0.072	10.69	197.2	-
		Obama	3.74	0.077	14.33	132.9	
(B):	CNN-based decoder	Biden	6.53	0.088	11.56	264.9	-
		Obama	4.23	0.066	12.65	125.9	
(C):	MLP + decoder	Biden	5.55	0.061	8.76	114.2	-
		Obama	3.58	0.060	9.37	103.4	
(D):	(C) + latent codes	Biden	5.68	0.062	**8.38**	**92.05**	-
		Obama	3.10	0.047	9.37	83.1	
(E):	(D) + $\mathcal{L}'_{rec}$	Biden	5.14	0.057	9.80	100.9	2.69
		Obama	**2.80**	0.041	9.97	82.1	2.31
(F):	(E) + gaze input (full model, Sec. 4.2)	Biden	**5.03**	**0.054**	9.42	107.2	**1.86**
		Obama	2.86	**0.041**	**9.33**	**79.3**	**2.18**

4.4 Discussion

Unlike NeRF-based models, our method is not 3D aware. Nonetheless, we claim that for the purposes of reenactment this is not an essential requirement for achieving consistent results in various head poses. Given that the training videos used by person-specific reenactment methods are captured from monocular and stationary cameras, they provide access only to a single view of the scene. NeRF-like models treat camera parameters coming from face tracking as different views of the scene. Although this choice provides multi-view consistency for the face region, it violates consistency for the upper torso and results in severe trembling. To our understanding, in order to aim for 3D consistency of the entire subject, one would need to build NeRF-based models using videos of faces captured by a moving camera from multiple viewpoints, as demonstrated most recently by RingNeRF [AXS+22]. For videos captured by stationary cameras, we have shown that our 2D model is able to generate frames with consistent appearance for different head poses of the target subject. Our experiments reveal that our approach surpasses 3D NeRF-based reenactment methods in visual quality. Future directions of our research include the exploration of person-generic alternatives to our model's architecture.

4.5 Summary

In this chapter, we presented Dynamic Neural Portraits, a novel method for video portrait synthesis and reenactment. In contrast to previous methods, our approach draws inspiration both from image-based neural rendering and neural scene representations. Our experiments demonstrate the superiority of our model in terms of visual quality and run-time performance, when compared to state-of-the-art video or audio-driven systems. We urge the reader to refer to the supplementary video, as our samples are best viewed as videos.

Part II

Person-Generic Methods for Head Animation and Editing

Chapter 5

HeadGAN: One-shot Head Synthesis and Editing

Video data synthesis [WLZ+18a, WLT+19], including talking head animation [WKZ18, ZSBL19, ZISL20, HKK+20, SLT+19a, SLT+19b, KGT+18] are particularly exciting and thriving research areas, with many applications. Over the past years, solutions were mainly given by the graphics community. For instance, Face2Face [TZS+16] performs face reenactment, by recovering facial expressions from a driving video and overwriting them to the target frames. More recent learning-based approaches [KGT+18], including Head2Head [KDRZ20, DKS+21], have sought to solve the problem of full head reenactment, which aims to transfer not only the expression, but also the head pose, from a source video to a target identity. The shortcoming of such algorithms is their dependence on long target videos, as they require training person-specific networks. At the same time, various methods have been proposed for reenacting human heads under a few-shot setting [WKZ18, WLT+19, ZISL20, HKK+20, SLT+19b, GSZ+18], where only a limited number of reference images of the target individual are available, even a single one. Many state-of-the-art approaches use facial key-points to guide synthesis [ZSBL19, ZISL20, ZISL20, SLT+19b], which usually leads to *identity preservation problems* during reenactment, as key-points encode appearance information. The problem becomes more prominent when the head geometry of the target identity is very different from the source.

In this chapter, we present **HeadGAN**, a novel one-shot GAN-based method for head animation and editing. We take a different approach from most existing one-shot or few-shot methods and condition synthesis using a 2D representation of 3D faces, known as PNCC images [ZLL+16]. We capitalise on prior knowledge of expression and identity disentanglement, enclosed within 3DMMs [BV99, PKA+09, BRP+18, CWZ+13, LBB+17]. Our decision to model faces with 3DMMs enables HeadGAN to operate as:

- a real-time *full head reenactment system*, operating at $\sim$ 20 fps
- an efficient method for video portrait *compression and reconstruction*
- a *facial expression editing method*
- a system for *novel head pose synthesis*
- a method for *face frontalisation*.



(a) Reference image (b) Reenactment (c) Reconstruction (self-reenactment)



(d) Expression Editing (e) Pose Editing (f) Frontalisation

Figure 5.1: Our proposed HeadGAN method performs reenactment (b), by fully transferring the facial expressions and head pose from a driving frame to a reference image. When the driving and reference identities coincide (c), it can be used for facial video compression and reconstruction. In addition, HeadGAN can be applied to facial expression editing (d), novel head pose synthesis (e), and face frontalisation (f).

Fig. 5.1 illustrates the tasks supported by our method. Apart from PNCCs, we optionally condition the generative process on speech features extracted from an input audio signal, which enables our method to perform accurate mouth synthesis, as suggested by our automated lipreading experiment. We perform extensive comparisons with state-of-the-art methods [WKZ18, WLT+19, ZISL20, SLT+19b, ZLL+20, RAP+21] and report superior image quality and performance on the tasks of one-shot facial video reconstruction and full head reenactment, as well as face frontalisation, even when compared to models [ZISL20] trained on the larger VoxCeleb2 [CNZ18] dataset. Finally, we conduct an ablation study in order to demonstrate the contribution of each component of our system.

5.1 Conditional Inputs

Our system conditions image synthesis on PNCC images, which is a face representation that can be extracted from any driving image or video with a 3D face reconstruction algorithm and adapted to the facial geometry of any reference image, disentangling identity-related facial shapes from expression deformations. We further improve the fidelity of synthesised mouth motions, by utilising audio features as a complementary input to our deep generative model.

5.1.1 Face Representation

In order to accurately transfer the expressions of the driving person while preserving the facial geometry of the target identity, we take advantage of prior knowledge of human faces, contained

within 3DMMs [BV99, PKA+09, BRP+18, CWZ+13, LBB+17]. Given that we have access to a single reference image of the target identity, the video-based 3D facial recovery algorithm we presented in Chapter 3 cannot be applied here. Moreover, the analysis-by-synthesis approach adopted by many state-of-the-art 3D face reconstruction algorithms [BAP+17, GPKZ19] estimates a lot of parameters (e.g. illumination, reflectance, shape, etc.) and solves a highly ill-posed problem, that is fitting 3DMMs to images based on texture information. On top of that, the majority of these methods require several seconds to recover the fitting parameters from one video frame. On the contrary, CNNs such as RetinaFace [DGV+20a] have shown incredible results on regressing dense 3D facial meshes from images at high speeds. Considering the above, we employ RetinaFace for regressing dense 3D keypoints (around 1K) from images, which is a neural network pre-trained on WIDER FACE dataset [YLLT16]. Then, we employ the key-point-based 3DMM fitting algorithm described in Section 2.1.2 to recover the shape and camera parameters of the tracked face. This step requires knowing the correspondence between the topology (vertices) of the 3DMM and the points estimated with RetinaFace. Prior to fitting, we use Procrustes Analysis to register the regressed 3D points with $\bar{\mathbf{x}}$, the mean shape of the 3DMM. The entire 3DMM fitting algorithm is presented below:

Algorithm 1: Key-point-based 3DMM fitting on facial images.

Input: Image: $\mathbf{y}$, 3DMM: $\{\mathbf{U}^{id}, \mathbf{U}^{exp}, \bar{\mathbf{x}}\}$
`/* Regress dense 3D facial key-points */`
$\mathbf{l} = \text{RetinaFace}(\mathbf{y})$
`/* Align key-points with the mean shape of 3DMM */`
$\mathbf{l}' = \text{Procrustes}(\bar{\mathbf{x}}, \mathbf{l})$
`/* Merge id. with exp. 3DMM */`
$\mathbf{U} = [\mathbf{U}^{id\top}; \mathbf{U}^{exp\top}]^{\top}$
`/* Compute Moore-Penrose inverse */`
$\mathbf{U}^{+} = (\mathbf{U}^{\top}\mathbf{U})^{-1}\mathbf{U}^{\top}$
`/* Recover shape parameters */`
$\mathbf{p} = \mathbf{U}^{+}(\mathbf{l}' - \bar{\mathbf{x}})$
`/* Compute affine camera matrix */`
$\mathbf{P} = \text{Least_squares}\ (\bar{\mathbf{x}}_{homog}, \mathbf{l}) \in \mathbb{R}^{3\times 4}$
`/* Recover camera parameters: scale, rotation, translation */`
$\mathbf{c} = \text{P_to_srt}(\mathbf{P})$
Result: shape parameters $\mathbf{p}$, camera parameters $\mathbf{c}$

Given a driving video of T frames, $\mathbf{y}_{1:T} = \{\mathbf{y}_t \mid t = 1, \ldots, T\}$, we apply the fitting algorithm to produce a sequence of *camera parameters* $\mathbf{c}_{1:T}$ and *shape parameters* $\mathbf{p}_{1:T}$, with $\mathbf{p}_t = (\mathbf{p}_t^{id}, \mathbf{p}_t^{exp})$. For each frame t, we obtain two types of shape parameters: identity related parameters $\mathbf{p}_t^{id} \in \mathbb{R}^{n_{id}}$, encoding facial geometry and expression parameters $\mathbf{p}_t^{exp} \in \mathbb{R}^{n_{exp}}$, representing facial deformations. This enables disentangling facial shape attributes that depend on identity from shape deformations caused by motion. Camera parameters indicate translation, rotation and orthographic scale. Additionally, given a reference image of the target identity $\mathbf{y}_{ref}$, we perform 3DMM fitting to obtain the target's shape parameters $\mathbf{p}_{ref}^{id}$, $\mathbf{p}_{ref}^{exp}$ and camera parameters $\mathbf{c}_{ref}$.

Next, for each frame t, we compute a "hybrid" 3D face $\mathbf{s}_t = [x_1, y_1, z_1, ..., x_N, y_N, z_N]^{\top} \in \mathbb{R}^{3N}$, as

$$\mathbf{s}_t = \bar{\mathbf{x}} + \mathbf{U}^{id}\mathbf{p}_{ref}^{id} + \mathbf{U}^{exp}\mathbf{p}_t^{exp}. \tag{5.1}$$

Here $\bar{\mathbf{x}} \in \mathbb{R}^{3N}$ is the mean shape, $\mathbf{U}_{id}$ is the identity orthonormal basis of LSFM morphable

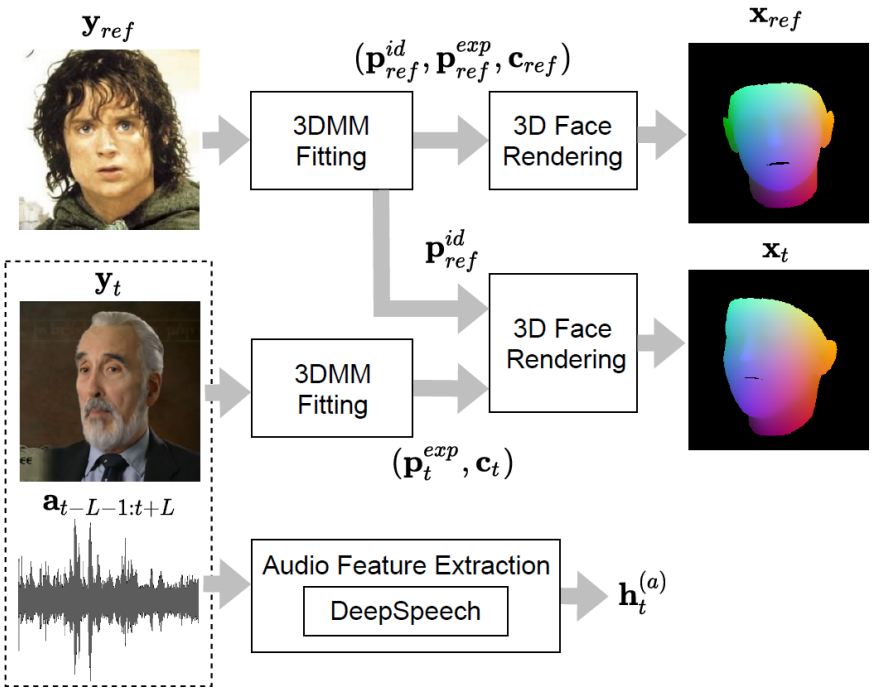


Figure 5.2: Data pre-processing stage. We recover and render the 3D face of the reference image $\mathbf{y}_{ref}$, as well as the driving frame $\mathbf{y}_t$, after adapting the identity parameters.

model [BRP+18], while $\mathbf{U}_{exp}$ is the expression orthonormal basis that has been built with the blendshapes of Facewarehouse [CWZ+13] by adopting Nonrigid ICP [CMZP17] in order to register the blendshapes model with LSFM model. By construction, the 3D shape $\mathbf{s}_t$ reflects the facial structure $\mathbf{p}^{id}_{ref}$ of the reference with the facial expressions $\mathbf{p}^{exp}_t$ of the driving identity. In this way, we address the target identity preservation problem.

Finally, we render the conditional face representation (PNCC image) $\mathbf{x}_t = \mathcal{R}(\mathbf{s}_t, \mathbf{c}_t)$, $t = 1, \ldots, T$, using the 3D shape $\mathbf{s}_t$ and camera parameters $\mathbf{c}_t$. We also render the PNCC $\mathbf{x}_{ref}$, which is the 3D face recovered from the reference image $\mathbf{y}_{ref}$, using camera parameters $\mathbf{c}_{ref}$ and shape $\mathbf{s}_{ref}$ that is computed with Eq. (5.1) using $\mathbf{p}^{id}_{ref}$ and $\mathbf{p}^{exp}_{ref}$ shape parameters. In Section 5.4.1 we discuss PNCC rendering in more detail. Summarising, given a reference image $\mathbf{y}_{ref}$ and a driving video $\mathbf{y}_{1:T}$, the data pre-processing pipeline recovers the PNCC image $\mathbf{x}_{ref}$, as well as a sequence of PNCCs, namely $\mathbf{x}_{1:T}$, which depict the 3D face extracted from the driver and adapted to the facial geometry of the target identity. These face representations are used to condition image synthesis.

5.1.2 Audio Features

As opposed to previous one-shot head reenactment systems, our method takes advantage of the driving audio stream and its correlation with facial and mouth movements. We split the driving audio signal into T parts $\mathbf{a}_{1:T}$, where each part $\mathbf{a}_t$ is aligned and corresponds to frame $\mathbf{y}_t$ of the driving video of length T. Then, we apply audio feature extraction to a window of $2L$ audio parts $\mathbf{a}_{t-L-1:t+L} = \{\mathbf{a}_{t-L-1}, \ldots, \mathbf{a}_t, \ldots, \mathbf{a}_{t+L}\}$, centred around frame t, to obtain a feature vector $\mathbf{h}^{(a)}_t$, which contains information from the past and future time steps. We employ [Gia15] for the extraction of low level features, such as MFCCs, signal energy and entropy, which yields

a feature vector $\mathbf{h}_t^{(a_L)} \in \mathbb{R}^{84}$. Then, we use DeepSpeech [HCC+14, AAA+16] for the extraction of character level logits from each part $\mathbf{a}_{t'} \in \mathbf{a}_{t-L-1:t+L}$. This results in $2L$ logits, which after concatenation gives a feature vector $\mathbf{h}_t^{(a_H)} \in \mathbb{R}^{2L \cdot 27}$. Our final audio feature vector for frame t is given as: $\mathbf{h}_t^{(a)} = [\mathbf{h}_t^{(a_L)^\top}; \mathbf{h}_t^{(a_H)^\top}]^\top \in \mathbb{R}^{300}$, for $L = 4$.

5.2 HeadGAN Learning Framework

Our GAN-based head reenactment system is equipped with a generator that is driven by two modalities: a representation of the 3D face (PNCC renderings) that is extracted from the driving video and the reference image, and optionally the audio features coming from the driving acoustic signal. Given $\mathbf{x}_{t-k:t}$, the PNCC image extracted from frame t, concatenated channel-wise with the PNCCs coming from the past $k = 2$ frames, the reference image $\mathbf{y}_{ref}$ with the corresponding PNCC representation $\mathbf{x}_{ref}$, as well as the audio feature vector $\mathbf{h}_t^{(a)}$, our generator hallucinates a photo-realistic image, given by

$$\tilde{\mathbf{y}}_t = G(\mathbf{x}_{t-k:t}, \mathbf{y}_{ref}, \mathbf{x}_{ref}, \mathbf{h}_t^{(a)}; \theta_G). \tag{5.2}$$

Conditioning synthesis on the spatio-temporal volume $\mathbf{x}_{t-k:t}$ helps to achieve temporal coherence across frames. The reference image $\mathbf{y}_{ref}$ provides information about the appearance of the target person (e.g. face texture, hair, torso, etc.), while audio features enhance the generative ability of network G across the face, and mainly the mouth area. In more detail, the generator consists of two sub-networks: *a dense flow network* F and a *a rendering network* R.

Dense flow network F The rendering network R relies on high quality visual features that reflect the appearance of the target identity. Nonetheless, we observed that using an encoder to extract such features from the reference image $\mathbf{y}_{ref}$, does not capitalise well on the potential of our rendering network's architecture. It has been proved more meaningful to align the visual feature maps with the desired head pose that appears in the driving PNCC representation $\mathbf{x}_t$. With this in mind, we propose a dense flow network, which learns to estimate an optical flow field $\mathbf{w}_t$ for warping the visual features. For that, we concatenate the reference image with the corresponding PNCC and pass the resulting $(\mathbf{y}_{ref}, \mathbf{x}_{ref})$ through an encoder, for the extraction of visual feature maps in three spatial scales $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}$, which represent the appearance of the target identity. Then, a decoder predicts the flow $\mathbf{w}_t$, guided by the driving PNCC representation $\mathbf{x}_{t-k:t}$, which is injected into F through SPADE [PLWZ19] blocks. Ideally, when applied on the reference image $\mathbf{y}_{ref}$, this dense flow should yield a warped image of the target person, with the same head pose and expression, as shown in the driving face representation $\mathbf{x}_t$. By applying the flow field on each visual feature map, we obtain the warped visual features $\bar{\mathbf{h}}_t^{(1)}, \bar{\mathbf{h}}_t^{(2)}, \bar{\mathbf{h}}_t^{(3)}$ and the warped reference image $\bar{\mathbf{y}}_t^{ref}$, all of which depend on the driving head pose at frame t. For an overview of the flow network F, please refer to Fig. 5.3. In more detail, the encoder is made up of three convolutional layers, each one equipped with instance normalization units [UVL16] and ReLU activation functions. The last two convolutions are performed with a stride of 2, for down-sampling the input twice. The decoder is equipped with SPADE blocks [PLWZ19], which are used to "inject" the driving face representation $\mathbf{x}_{t-k:t}$ as modulation input. Here we down-sample $\mathbf{x}_{t-k:t}$ to match it with the spatial size of each SPADE layer, similarly to the seminal SPADE-based decoder [PLWZ19]. We employ two Pixel Shuffle [SCH+16] layers for up-sampling. Finally, the predicted flow is calculated with a convolutional

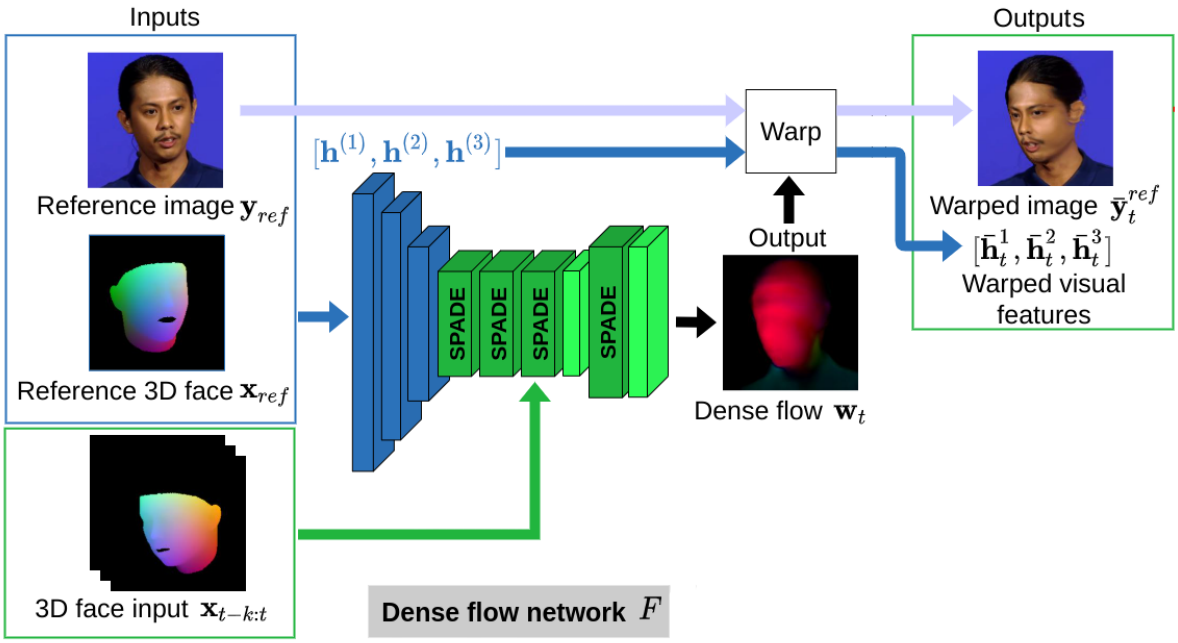


Figure 5.3: Overview of the dense flow network F. This module computes a flow field for warping the reference image and features, according to its 3D face inputs (PNCC representations).

output layer. The architecture of F is presented in Table 5.1.

Table 5.1: The architecture of dense flow network F.

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

Rendering network R At the core of our generator, the rendering network aims to translate the PNCC representation $\mathbf{x}_{t-k:t}$ into a photo-realistic image $\tilde{\mathbf{y}}_t$ of the target face. This is achieved with the assistance of high quality audio features $\mathbf{h}_t^{(a)}$ and visual feature maps $\bar{\mathbf{h}}_t^{(1)}, \bar{\mathbf{h}}_t^{(2)}, \bar{\mathbf{h}}_t^{(3)}$. First, an encoder receives $\mathbf{x}_{t-k:t}$ as input and applies a sequence of convolutional layers with down-sampling. This part of the network has a similar structure to the encoder of F. Then, a decoder consisting of alternating SPADE [PLWZ19] and AdaIN [HB17] layers generates the desired frame $\tilde{\mathbf{y}}_t$. These adaptive normalisation layers enable injecting 2D visual

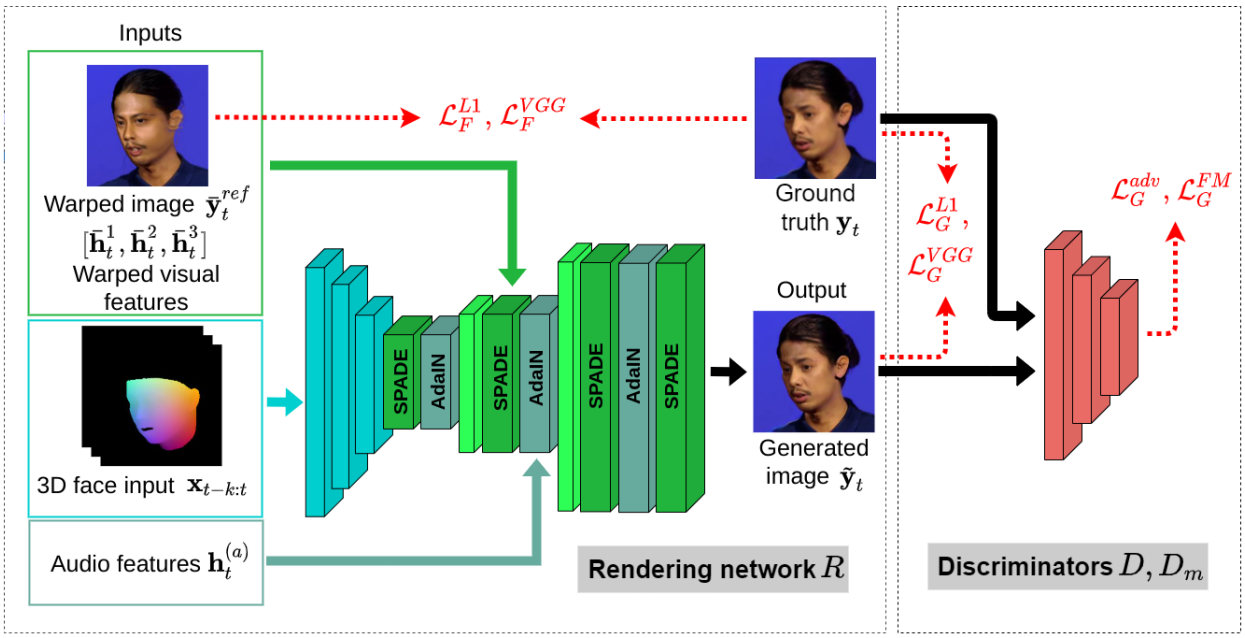
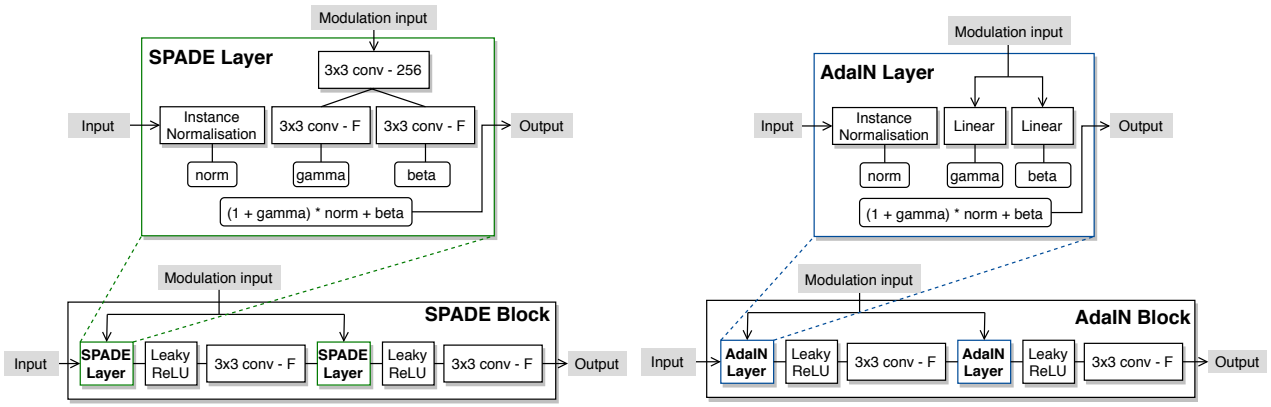


Figure 5.4: Overview of the rendering network R. This network uses the visual information of the warped reference image and warped features along with the audio features in order to translate the driving PNCC input into a photo-realistic image of the reference identity.

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

Table 5.2: The architecture of rendering network R.

feature maps into the rendering network through SPADE blocks, as well as 1D audio features through AdaIN blocks. As opposed to the original SPADE-based decoder [PLWZ19], where the modulation input of all SPADE layers is the same segmentation map down-sampled to match the spatial size of each layer, we capitalise on visual feature maps of multiple spatial scales $\bar{\mathbf{h}}_t^{(1)}, \bar{\mathbf{h}}_t^{(2)}, \bar{\mathbf{h}}_t^{(3)}$, $\bar{\mathbf{y}}_t^{ref}$ as modulation inputs to SPADE blocks. Nonetheless, we pass the same audio feature vector $\mathbf{h}_t^{(a)}$ to AdaIN blocks of all spatial scales. Our decoder is further



(a) SPADE Block architecture. (b) AdaIN Block architecture

Figure 5.5: Our SPADE and AdaIN blocks are based on the SPADE Resnet blocks proposed in [PLWZ19] without a residual component, as we always keep the same number of input channels F at the output, both on SPADE and AdaIN blocks.

equipped with Pixel Shuffle layers [SCH+16] for up-sampling, which contribute to the quality of images. After the last decoding block, a convolutional layer computes the synthetic image. The structure of our rendering network is presented in Fig. 5.4 and Table 5.2.

Discriminators D and D_m Our generator is trained along with a discriminator, which receives the synthetic $(\mathbf{x}_t, \tilde{\mathbf{y}}_t)$ or real pair $(\mathbf{x}_t, \mathbf{y}_t)$ and learns to distinguish between them. We use a second discriminator D_m, which focuses on the mouth region. Apart from the real $\mathbf{y}_t^m$ or generated cropped mouth area $\tilde{\mathbf{y}}_t^m$, this network is conditioned on the audio feature vector $\mathbf{h}_t^{(a)}$, which is spatially replicated and then concatenated to the cropped images channel-wise. Both D and D_m have a similar architecture to the discriminator presented in [PLWZ19]. We apply Spectral Normalisation [MKKY18] to all normalisation layers of the two discriminators.

Training Objective. Networks F and R which constitute the generator are optimised jointly. We train HeadGAN on reconstruction, by applying perceptual and pixel losses both on the warped and generated images, as seen in Fig. 5.4 with red arrows. An adversarial Hinge loss [LY17] combined with a feature matching loss [WLZ+18b] further increase the photo-realism of results. The adversarial loss term for G is given as

$$\mathcal{L}_G^{adv} = -\mathrm{E}_{p_{data}}[D(\mathbf{x}_t, \tilde{\mathbf{y}}_t) + D_m(\mathbf{h}_t^{(a)}, \tilde{\mathbf{y}}_t^m)], \tag{5.3}$$

where $\mathbf{x}_t$ is the driving PNCC input, $\mathbf{h}_t^{(a)}$ is the input audio feature vector, $\tilde{\mathbf{y}}_t$ is the synthetic frame generated by G and $\tilde{\mathbf{y}}_t^{mouth}$ is the corresponding cropped mouth area of size 64×64. The image discriminator D is optimised by minimising the loss

$$\begin{aligned} \mathcal{L}_D^{adv} = -\mathrm{E}_{p_{data}}[&\min(0, -1 + D(\mathbf{x}_t, \mathbf{y}_t) \\ &- \min(0, -1 - D(\mathbf{x}_t, \tilde{\mathbf{y}}_t)]. \end{aligned} \tag{5.4}$$

and the mouth discriminator D_m with the corresponding loss

$$\begin{aligned} \mathcal{L}_{D_m}^{adv} = -\mathrm{E}_{p_{data}}[&\min(0, -1 + D_m(\mathbf{x}_t, \mathbf{y}_t^m) \\ &- \min(0, -1 - D_m(\mathbf{x}_t, \tilde{\mathbf{y}}_t^m)]. \end{aligned} \tag{5.5}$$

Since during training we perform self-reenactment, we have access to the ground truth frame $\mathbf{y}_t$. Thus, the generative network G is trained by minimising also a reconstruction loss term between the generated and ground frames, in the image pixel space

$$\mathcal{L}_G^{L1} = \mathrm{E}_{p_{data}}[||\tilde{\mathbf{y}}_t - \mathbf{y}_t||_1], \tag{5.6}$$

as well as in a feature space, using feature maps extracted by a pre-trained VGG network [SZ14]:

$$\mathcal{L}_G^{VGG} = \mathrm{E}_{p_{data}}[\sum_l ||VGG_l(\tilde{\mathbf{y}}_t) - VGG_l(\mathbf{y}_t)||_1]. \tag{5.7}$$

Similarly to VGG loss, we use the two discriminators to compute visual features, both from real and synthetic frames, and compute a feature matching loss $\mathcal{L}_G^{FM}$ that was originally proposed for training pix2pixHD [WLZ+18b], as it has been proven very effective at increasing the photo-realism of generated samples.

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

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

$$\begin{aligned}\mathcal{L}_G = \mathcal{L}_G^{adv} + \lambda_{L1}\mathcal{L}_G^{L1} + \lambda_{VGG}\mathcal{L}_G^{VGG} + \lambda_{FM}\mathcal{L}_G^{FM} \\ + \lambda_{L1}\mathcal{L}_F^{L1} + \lambda_{VGG}\mathcal{L}_F^{VGG}.\end{aligned} \tag{5.8}$$

We set $\lambda_{L1} = 50$ and $\lambda_{VGG} = \lambda_{FM} = 10$.

5.3 Advantages of 3D face modeling

In Section 5.2, we explained how network F learns to predict a flow field, in order to warp and align the reference image and its visual features to the desired driving pose. The *semantic information* enclosed within the PNCC face representation allows our dense flow network F to learn a precise flow of the facial region, as it provides a dense correspondence of facial points, between the reference and driving images. Compared to scene flow [LQG19, KRZ20] that can be obtained from 3D meshes, our flow field hallucinated by F exists in areas where 3D representation is missing, such as hair and upper body, where warping is equally important. Furthermore, as 3DMMs allow to disentangle identity from expression, our choice to condition the rendering network R on a representation of the 3D face that is extracted from the driving frames and adapted to the identity characteristics of the target enables HeadGAN to tackle the identity preservation problem during cross-identity reenactment.

Additionally, modeling faces with 3DMMs makes HeadGAN very efficient for video conferencing applications. It enables a *"sender"* to efficiently compress a driving frame $\mathbf{y}_t$, in the form of expression parameters $\mathbf{p}_t^{exp} \in \mathbb{R}^{28}$ and camera parameters $\mathbf{c}_t \in \mathbb{R}^7$, giving a total of 35 floating point values. Then, these parameters can be used by a *"receiver"* to render the PNCC representation of the 3D face and use the generator to reconstruct the driving frame. A single reference image needs to be sent once, in the beginning of the session.

5.4 Experimental Evaluation

5.4.1 Implementation details

3D Face Rendering Given a set of camera parameters $\mathbf{c}$ and a 3D facial shape $\mathbf{s} \in \mathbb{R}^{3N}$ (see Eq. (5.1)), we rasterise the 3D mesh and produce a visibility mask $I \in \mathbb{R}^{H \times W}$ on the image plane. Each spatial location (pixel) of I stores the index of the corresponding visible triangle of the 3D face. Then, we use the mean shape $\bar{\mathbf{x}}$ of the 3DMM, in order to find the normalised x-y-z coordinates of the center of each visible triangle. In this way we obtain a PNCC image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, where each pixel contains three coordinates. These values can be interpreted as colors and therefore texture of the 3D face [ZLL+16].

Dataset and Training We train and evaluate HeadGAN on VoxCeleb [NCZ17] dataset, which contains over 100,000 videos of 1,251 identities, at 256×256 resolution. We maintain the original train and test split. As a pre-processing step, we compute a PNCC image for each video frame in the dataset and extract per-frame audio feature vectors. During training, we perform self-reenactent, as we randomly sample the reference image from the driving video. This provides access to ground truth data, enabling us to use reconstruction loss terms and train the generator. For the optimisation of HeadGAN, we use the ADAM [KB14] with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and learning rate $\eta = 0.0002$, both for the generator and discriminators.

5.4.2 Evaluation Metrics

We quantitatively compare HeadGAN with baseline methods, using the metrics described below:

L1 distance (L1) We evaluate the reconstructive ability of models by computing the mean $l1$-distance, between the synthesised and ground truth frames. We average the distance across channels, pixel locations and frames in the test set, to obtain the L1 metric. Please note that RGB channels are in the range [0, 255].

Peak signal-to-noise ratio (PSNR) This is another metric to measure the quality of reconstructed videos. PSNR is the ratio between the maximum possible power of a signal and the power of noise that affects the fidelity of its representation, defined as: $20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10} MSE$. Here, $MAX_I = 255$ and MSE denotes the mean squared error, computed across color channels, spatial locations and frames. PSNR is expressed in dBs.

Learned Perceptual Image Patch Similarity (LPIPS) Metrics such as L1 and PSNR are simple shallow functions that are not able to account for many nuances of human perception. On the contrary, LPIPS [ZIE+18] is a metric that adopts a neural network to extract features from images. This network has been trained to solve challenging visual prediction tasks and has learned a representation that correlates well with perceptual judgments. Then, a similarity score between two images is calculated based on their corresponding visual features.

Fréchet Inception Distance (FID) We employ FID [HRU+17, Sei20] as a measure of similarity between the dataset of real images and the dataset of images generated by the models. This score provides a useful insight into the photo-realism of synthetic frames.

Fréchet Video Distance (FVD) Given that we handle video data, it is important to evaluate the generative performance of models using a metric which takes into account the temporal coherence between frames. To that end, we calculate the FVD score [UvSK+18] of generated sequences, which has shown to correlate well with the human judgment on visual quality of generated videos.

Cosine Similarity (CSIM) Cosine similarity is a widely-used metric, which measures identity preservation in synthetic frames. We use ArcFace [DGXZ19] as an identity recognition network, in order to compute pairs of embedding vectors from the driving and corresponding generated images. Then, we calculate the cosine similarity between all pairs of embedding vectors in the dataset and report its average value. During reenactment, where no ground truth images are available, we extract the embedding vector from the reference image and compare it with the embeddings coming from synthetic frames. This leads to smaller CSIM values in reenactment, as the pose of the reference image is differs from the one in the generated images, and the identity recognition network's output is not completely unaffected by head pose.

Action Units Hamming distance (AU-H) In order to measure the expression transferability of models, we use OpenFace [ALS16] and more specifically [BMR15], for the detection of Action Units (AU) in driving and generated images. Facial Action Coding System (FACS) is a system to taxonomise human facial movements by their appearance. Using FACS it is possible to code nearly any anatomically possible expression, deconstructing it into the specific AUs that produced the expression. It is a common standard to objectively describe facial expressions. We use OpenFace to recognise if a set of AUs is present in a facial image or not, calculating a boolean vector of AUs. Then, we compute the Hamming distance $\in [0, 1]$ between AU boolean vectors, extracted from the corresponding driving and synthetic frames, and average across all frames.

Average Rotation Distance (ARD) This metric evaluates pose transfer. We use the camera parameters from 3D face reconstruction to compute the Euler angles that correspond to head poses in the driving and generated frames. Then, the average L1-distance of Euler angles across all frames is determined, in terms of degrees.

Average Rotation Error (ARE) This is a metric similar to ARD, but measures the average L1-distance of Euler angles from the frontal pose (zero degrees), across all images.

5.4.3 Comparison with baselines

We perform extensive comparisons with state-of-the-art methods [WKZ18, WLT+19, ZISL20, SLT+19b, ZLL+20, RAP+21] and report superior image quality and performance, in terms of standard GAN metrics [HRU+17, UvSK+18] on the tasks of reconstruction, reenactment and frontalisation, even when compared to models [ZISL20] trained on the larger VoxCeleb2 [CNZ18] dataset. Lastly, we conduct an ablation study in order to demonstrate the contribution of key components of our system.

Reconstruction (Self-reenactment) First, we compare our approach with four state-of-the-art methods on the problem of self-reenactment, where the reference identity coincides with the driving one. Here, the task for HeadGAN is to reconstruct the driving video from the 3D face representation sequence, using a single reference image to access appearance information.



Figure 5.6: Qualitative comparison of HeadGAN with four baseline methods on the task of reconstruction (self-reenactment).

We perform both qualitative and quantitative comparisons with X2Face [WKZ18], few-shot vid2vid [WLT+19], Bi-layer Neural Avatars [ZISL20] and First Order Motion Model [SLT+19b]. For [WKZ18] and [SLT+19b], we use the pre-trained models provided by the authors, trained on VoxCeleb. We trained [WLT+19] from scratch, since no checkpoint was available. Lastly, we used the model provided by the authors of [ZISL20], trained on the larger VoxCeleb2* [CNZ18].

Table 5.3: Quantitative results on the tasks of reconstruction for VoxCeleb [NCZ17] test set. For all metrics, except from PSNR and CSIM, lower is better.

	Reconstruction					
Method	L1 (↓)	PSNR (↑)	LPIPS (↓)	FID (↓)	FVD (↓)	CSIM (↑)
X2Face [WKZ18]	13.49	20.69	0.260	130.2	697	0.600
fs-vid2vid [WLT+19]	17.15	18.52	0.197	62.8	471	0.542
Bi-layer* [ZISL20]	*12.18*	20.19	*0.152*	92.2	394	0.590
FOMM [SLT+19b]	12.34	*20.93*	0.153	*64.9*	*338*	*0.754*
HeadGAN	**11.32**	**21.46**	**0.112**	**36.1**	**254**	**0.807**

The samples illustrated in Fig. 5.6 show that our method generates far more realistic images than the baselines with better preserved appearance traits. For the numerical evaluation of reconstruction we use L1 distance between the generated and ground truth frames, as well as Peak signal-to-noise ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS) [ZIE+18]. We access the realism of frames with Fréchet Inception Distance (FID) [HRU+17] and Fréchet Video Distance (FVD) [UvSK+18] metrics. We use Cosine Similarity (CSIM) [DGXZ19] to mea-

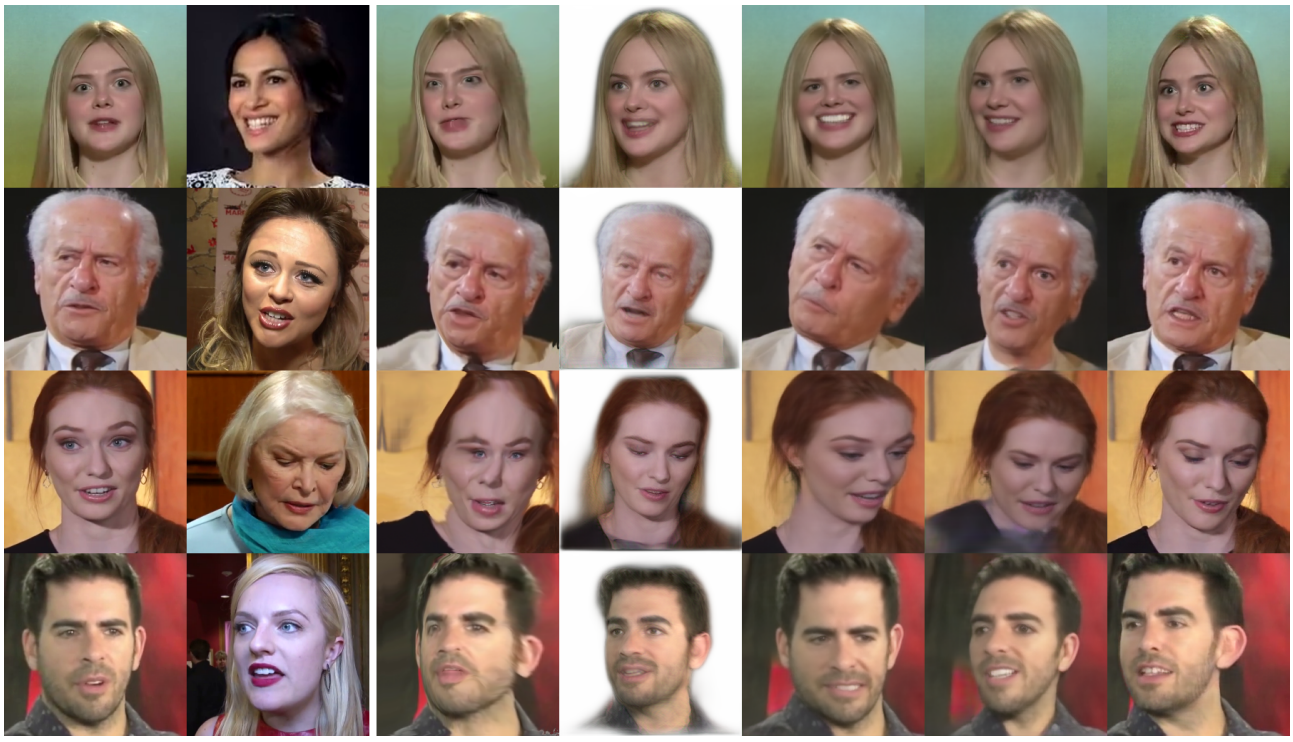


Figure 5.7: Qualitative comparison with baselines, on the task of reenactment. FOMM-Rel refers to relative key-point coordinates strategy and FOMM-Abs to absolute key-point coordinates, used in First Order Motion Model [SLT+19b].

sure identity preservation. The results are presented in Table 5.3 and reveal that HeadGAN outperforms all four baselines by a noteworthy margin, in every single metric.

Reenactment The objective of reenactment is to fully transfer the head pose and facial expressions from a driving sequence to the person shown in the reference image, while preserving the latter identity, as the driving and reference subjects are now different. To that end, we selected 15 random (video, image) pairs from VoxCeleb test set and performed reenactment, generating around 6K frames in total with each method. Apart from image quality (FID) and identity preservation (CSIM), we further evaluate the pose and expression transferability of systems, with Average Rotation Distance (ARD) in degrees and Action Units Hamming distance (AU-H) [BMR15], respectively. As can be seen in Table 5.4, HeadGAN creates superior

Table 5.4: Quantitative results on the task of reenactment for VoxCeleb [NCZ17] test set. For all metrics, except from CSIM, lower is better.

	Reenactment			
Method	FID (↓)	CSIM (↑)	ARD (↓)	AU-H (↓)
X2Face [WKZ18]	122.1	0.520	4.39	0.346
Bi-layer* [ZISL20]	172.8	0.563	**1.01**	**0.296**
FOMM [SLT+19b]	*63.7*	**0.765**	12.53	0.400
HeadGAN	**58.0**	*0.688*	*1.35*	*0.326*

samples in terms of visual quality. Bi-layer Neural Avatars [ZISL20] performs slightly better on the task of facial expression transfer, which could be attributed to the fact that it was trained on the larger VoxCeleb2 [CNZ18]. However, it performs poorly on identity preservation, as it conditions synthesis on facial landmarks, which unavoidably pass on identity related information from the driving subject. On the other hand, FOMM [SLT+19b] uses relative key-point locations to account for the identity preservation problem that seems to increase CSIM. Nonetheless, this comes at the expense of pose transfer, as the model requires the face in the first frame of the driving video to have the same pose with the reference face, which is very rarely the case, also confirmed by the large ARD. When FOMM uses absolute key-point locations instead, in order to accurately transfer pose, the identity preservation problem becomes apparent and CSIM drops to 0.587. This behaviour can be observed visually in Fig. 5.7, where the head geometry of the driver is reflected in the generated samples of FOMM-abs. Different than the baselines, HeadGAN performs well on all three requirements of successful reenactment (pose and expression transfer, as well as identity preservation). Please note that we have omitted comparisons with MarioNETte [HKK+20] and Warp-guided GANs [GSZ+18] since their source code is not publicly available.

Frontalisation Our choice to use a 3D face rendering to condition synthesis allows us to set the desired head pose manually, without the need of driving footage. By re-setting the camera parameters to frontal, we are able to generate a frontal view of the reference image. We compare HeadGAN on the task of frontalisation with pixel2style2pixel (psp) [RAP+21] and Rotate-and-Render (RaR) [ZLL+20]. For that, we randomly select one frame from each video of VoxCeleb test split and frontalise it. Since there is no audio input for this task, we trained a variation of HeadGAN without AdaIN layers for audio features. Then, we measure the photo-realism of generated samples with FID, the identity preservation with CSIM, and finally the Average Rotation Error (ARE) as the deviation from the frontal pose, in degrees. We define the frontal pose as zero Euler angles in camera rotation parameters that come from 3DMM fitting. Therefore, ARE is only used as a reference here, for a sanity check. The results are displayed in Table 5.5. HeadGAN performs equally well on CSIM with RaR [ZLL+20] and surpasses baselines on image quality and frontalisation accuracy. Please see Fig. 5.8 for a visual comparison.



Figure 5.8: Qualitative comparison with psp and RaR methods on face frontalisation.

Table 5.5: Quantitative results on frontalisation.

Method	FID (↓)	CSIM (↑)	ARE (↓)
psp [RAP+21]	147.8	0.130	2.66
RaR [ZLL+20]	*88.4*	*0.753*	*2.65*
HeadGAN	**30.1**	**0.766**	**0.76**

5.4.4 Editing the Expression and Pose of Image Portraits

Our model can be further used as an image editing tool. Given a reference image $\mathbf{y}_{ref}$ and the estimated shape and camera parameters $\mathbf{p}^{id}_{ref}$, $\mathbf{p}^{exp}_{ref}$ and $\mathbf{c}_{ref}$, first we render the PNCC face representation $\mathbf{x}_{ref}$. Then, we re-adjust the expression or camera parameters manually, compute a pseudo-driving 3D face and render the corresponding PNCC image $\mathbf{x}_t$. We pass $\mathbf{x}_t$, $\mathbf{y}_{ref}$, $\mathbf{x}_{ref}$ through the generator to obtain a synthetic image with the novel expression or pose, reflecting the adjusted parameters. In Fig. 5.9 we show the results after manipulating the first three principal components (p. c.) of expression, as well as the camera angles.



Figure 5.9: Expression and pose editing with HeadGAN. Please note that we use our model's variation without audio input (AdaIN blocks) for this task.

5.4.5 Ablation Study

We conducted an ablation study in order to assess the significance of the dense flow network F, the advantage of the 3D face representation compared to sketches of sparse landmarks, which is a widely-used face representation [WLT+19, ZSBL19, ZISL20], and the contribution of audio modality. In order to evaluate the importance of network F, we removed its decoding layers that predict optical flow. We kept its encoder, which extracts appearance feature maps from the reference image. Then, instead of warping the features maps and reference image, we passed them directly to the SPADE layers of the rendering network R. In addition, we implemented a HeadGAN variation with landmarks, by conditioning synthesis on 2D image sketches, drawn by connecting 2D landmarks with edges [WLT+19]. As can be seen in Table 5.6, the full model outperforms all variations.

Table 5.6: Ablation study numerical results. Please note that due to time constraints, we trained all models for half epochs in this experiment.

Method	FID (↓)	FVD (↓)	CSIM (↑)
HeadGAN w/o network F	63.3	473	0.307
HeadGAN w/ landmarks	55.7	371	0.699
HeadGAN w/o audio input	55.1	356	0.687
HeadGAN	**50.9**	**334**	**0.716**

As suggested by all metrics, the dense flow network F is an essential component of our system. In Fig. 5.10 we visualise the importance of warping the reference image and feature maps in the quality of synthesised samples. Finally, the use of PNCCs (instead of landmarks) alleviates the identity preservation problem, which can be better understood visually, in Fig. 5.11.



Reference Driving HeadGAN w/o network F HeadGAN

Figure 5.10: The significance of dense flow network F in the quality of images.



Reference Driving HeadGAN w/ landmarks HeadGAN

Figure 5.11: The identity preservation problem becomes prominent when conditioning image synthesis on sketches of facial landmarks, instead of PNCC images.

Lipreading experiment We further evaluate the contribution of audio inputs to our system quantitatively, by employing an external lipreading network to classify synthetic videos. To that end, we chose 25 word classes of BBC dataset [CZ16] and trained a lipreading classifier [ST17] to differentiate between these classes, using the default training split. After that, we reconstructed the test split with the same 25 words of BBC dataset, using a random frame from each video as reference. We report a lipreading accuracy of 97% on real test samples, 82% on samples generated using the the full model and 73% on synthetic data produced by the variation without considering audio input. Our findings suggest that the audio modality contributes largely on the generation of more plausible lip movements.

5.5 Summary

In this chapter, we presented HeadGAN, a novel one-shot method for animating heads, driven by renderings of 3D faces and audio features. Compared to previous state-of-the-art methods for few-shot talking head synthesis and motion transfer, our framework exhibits superior reenactment performance and better photo-realism. Our method can be further used for video portrait compression and reconstruction, pose and expression editing, as well as face frontalisation. We urge the reader to refer to our supplementary video for a better visualisation of our synthesised samples. More visual results are also presented in Appendix A.

Chapter 6

Reenactment with Canonical Keypoints & Explicit Gaze Control

The design choices regarding face representation are important aspects of one-shot talking head synthesis. Sketches drawn from sparse 2D facial landmarks (usually 68 key-points) is a common representation of faces [ZSBL19, WLT+19, ZISL20]. More recent methods, such as [HKK+20] and [GSZ+18] use 3D landmarks instead. Among other benefits, the selection of a three-dimensional representation enables explicit *head pose control*, as it allows to intervene in the pose to be generated, by rotating the 3D points. Dense 3D facial meshes combined with statistical priors of the face, such as 3D Morphable Models (3DMMs) [BV99, PKA+09, BRP+18, CWZ+13, LBB+17] are other widely used facial representations, which have been recently adopted for reenacting heads [KGT+18, KDRZ20, DZS21]. This modeling enables to disentangle shape (identity) from expression-related deformations, and therefore helps to tackle the *identity preservation problem* in cross-identity motion transfer, which is neither straightforward nor trivial with sparse landmarks. In a quite different line of work, [SLT+19a, SLT+19b] attempt to solve the more general problem of motion transfer for arbitrary objects, proposing a *model-free* method based on unsupervised 2D key-point detection. Following up, face-vid2vid [WML21] proposes a model that focuses on faces and learns to detect *canonical* 3D key-points without supervision, as a way to disengage identity from expression and pose.

Given that they deal with motion transfer, most recent person-generic systems are flow-based [HKK+20, GSZ+18, WKZ18, SLT+19a, SLT+19b, WML21, DZS21], meaning that they make use of optical flow to warp the reference image(s) or visual features according to the driving facial pose and expression, as a first step prior to image synthesis. The aforementioned model-free approaches [SLT+19a, SLT+19b, WML21] are able to infer important key-points for motion, as they approximate optical flow in areas near these points. For example, [SLT+19b] and [WML21] use a *first order approximation of motion*. On the contrary, methods such as [WKZ18, HKK+20, DZS21] learn a *dense optical flow* for all pixels, without resorting to approximations.

In this chapter, we propose **Free-HeadGAN**, a system based on the architecture of HeadGAN [DZS21] but "free" from statistical priors of faces, like 3DMMs. In order to overcome the identity preservation obstacles in reenactment and inspired from [WML21], we propose a network that regresses 3D key-points along with head pose and expression deformations. Different from [WML21] our method does not predict canonical key-points directly. Instead, we compute them based on the regressed 3D points, after removing the estimated pose and expression in-

formation. We supervise our key-point detector with pseudo-ground truth 3D facial landmarks extracted with RetinaFace [DGV+20a]. In this way, we ensure that all key-points are always placed on meaningful parts of the face, avoiding cases where no key-points are assigned to important regions such as the eyes and mouth, as it commonly happens with unsupervised models [SLT+19a, SLT+19b, WML21]. In addition, we propose a gaze estimation network that makes explicit eye gaze transfer possible. Lastly, contrary to [SLT+19b] and [WML21], we do not use a first order approximation of motion, but rather use a neural network that learns to predict a dense optical flow, similarly to HeadGAN [DZS21]. The contributions presented in the chapter can be summarised in the following:

1. We release HeadGAN [DZS21] from 3DMM priors, using sparse 3D landmarks. As suggested by our qualitative and quantitative experiments, we achieve comparable and in many cases improved results.

2. We design a module that performs disentanglement of identity, expression and pose through the computation of canonical 3D key-points.

3. We train a network that regresses 3D meshes of the eyes. We use these meshes to obtain the direction of gaze, which is then used to condition image synthesis. To the best of our knowledge, we deliver the first person-generic reenactment system with explicit gaze control.

4. We show a N-shot extension of our framework.

6.1 Free-HeadGAN model

Our proposed talking head synthesis system consists of three discrete networks: a network for predicting 3D key-points, head pose information and expression deformations for the computation canonical key-points (Section 6.1.1), a gaze estimation network (Section 6.1.2) and an image generator (Section 6.1.3). Each component of Free-HeadGAN is trained separately on its individual task.

6.1.1 Computation of Canonical Key-points

Given a portrait image, our goal is to extract a set of key-points in a canonical space that are independent both from the head pose and facial expressions of the subject. This representation merely depends on the geometry of the input face and therefore encodes only identity-related information. To that end, we propose a neural network E_{can} that learns to a) regress sparse 3D facial key-points $\mathbf{p} = \{\mathbf{p}_k\}_{k=1,\dots,K}$, with $\mathbf{p}_k \in [-1, 1]^3$, b) estimate head pose, as an affine transformation $\mathcal{T} = \{s, \mathbf{R}, \mathbf{t}\}$ and c) compute a 3D vector $\mathbf{d}_k$ for each key-point k, that models the non-linearity of deformations caused by facial expressions. The architecture of E_{can} follows that of the head pose and expression deformation estimator in [WML21], with the addition of one more affine output layer that predicts $\mathbf{p}$, K key-points in number.

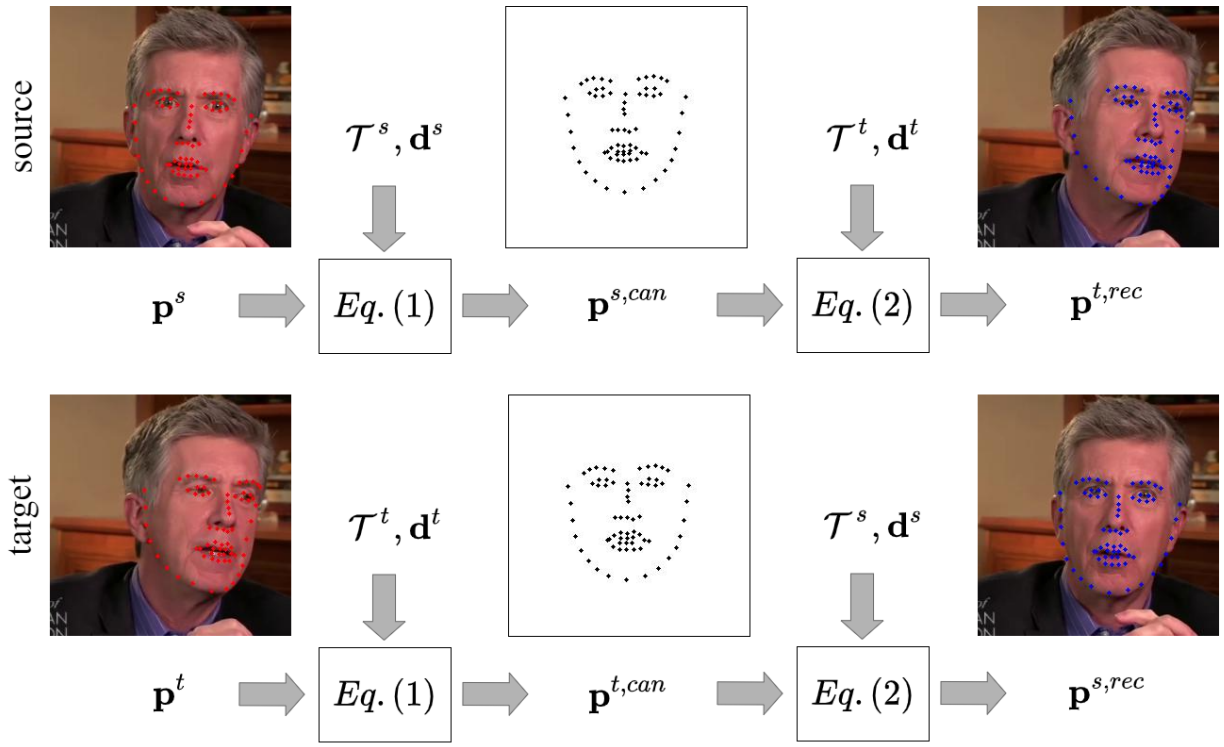


Figure 6.1: The pipeline for computing the canonical key-points during training. Using a pair of frames coming from the same video, we train E_{can} to disentangle identity from expression and pose. We project the regressed 3D key-points into the canonical space and then try to reconstruct them, after swapping the canonical points of the source and target images.

The canonical key-points are obtained by first estimating the 3D points and then subtracting the expression deformations and removing translation, rotation, and scale:

$$\mathbf{p}_k^{can} = \frac{1}{s}\mathbf{R}^{-1}(\mathbf{p}_k - \mathbf{d}_k - \mathbf{t}), \quad k = 1, \dots, K. \tag{6.1}$$

We also define the inverse operation, which brings canonical points back to the original 3D space, by adding scale, rotation, translation, as well as the expression deformation:

$$\mathbf{p}_k = s\mathbf{R}\mathbf{p}_k^{can} + \mathbf{t} + \mathbf{d}_k, \quad k = 1, \dots, K. \tag{6.2}$$

We observed that predicting the expression-related deformations $\mathbf{d}_k$ in the original 3D space, as in [WML21], yields significantly better results than placing the deformation in the canonical space.

In order to train network E_{can}, we are given a source and target image pair $(\mathbf{x}^s, \mathbf{x}^t)$, which depicts the same person at two different poses and expressions. We use E_{can} to regress 3D key-points as well as pose transformations and expression perturbations from both images: $\mathbf{p}^s, \mathcal{T}^s, \mathbf{d}^s$ and $\mathbf{p}^t, \mathcal{T}^t, \mathbf{d}^t$. After that, we apply Eq. (6.1) to get the corresponding canonical points $\mathbf{p}^{s,can}$ and $\mathbf{p}^{t,can}$, as shown in Fig. 6.1. Then, based on Eq. (6.2) we aim to recover the target key-points, by transforming the canonical representation of the source using the target pose and expression, and vice versa, which results in points $\mathbf{p}^{s,rec}$ and $\mathbf{p}^{t,rec}$. Having access to pairs of images of the same identity is crucial for guiding E_{can} to focus on expression and pose discrepancies between images, and therefore not include identity-related information in

the deformation vectors $\mathbf{d}^s$ and $\mathbf{d}^t$.

We optimise E_{can} in a supervised manner, using $K = 68$ 3D facial landmarks estimated by a pre-trained RetinaFace [DGV+20a] model, as pseudo-ground truth data. That is, given the ground truth source and target landmarks $(\mathbf{l}^s, \mathbf{l}^t)$, we minimise the distance

$$\mathcal{L}^p = ||\mathbf{p}^s - \mathbf{l}^s||_2^2 + ||\mathbf{p}^t - \mathbf{l}^t||_2^2, \tag{6.3}$$

which forces E_{can} to predict accurate facial key-points. We noticed that learning to estimate 3D key-points helps E_{can} to predict the perturbations caused due to expressions more efficiently. We further minimise the key-point reconstruction distance

$$\mathcal{L}^{rec} = ||\mathbf{p}^{s,rec} - \mathbf{l}^s||_2^2 + ||\mathbf{p}^{t,rec} - \mathbf{l}^t||_2^2, \tag{6.4}$$

which forces E_{can} to learn the affine pose transformations as well as the deformation vectors, in an effort to recover the source and target key-points. In order to assist E_{can} distinguish between rigid and non-rigid transformations, we penalise the error between the predicted target head rotation $\mathbf{R}^t$ and the corresponding ground truth $\mathbf{R}^t_*$ (here expressed as Euler angles). Furthermore, a regularisation term on the expression deformation vectors ensures that key-point perturbations due to expressions are kept small, as we want to avoid encoding identity-specific details in these vectors:

$$\mathcal{L}^R = ||\mathbf{R}^t - \mathbf{R}^t_*||_2^2, \quad \mathcal{L}^d = ||\mathbf{d}^s||_2^2 + ||\mathbf{d}^t||_2^2. \tag{6.5}$$

Combining the loss terms defined above, the overall objective function for network E_{can} is given as

$$\mathcal{L}_{E_{can}} = \lambda_p \mathcal{L}^p + \lambda_{rec} \mathcal{L}^{rec} + \lambda_R \mathcal{L}^R + \lambda_d \mathcal{L}^d \tag{6.6}$$

with the hyper-parameters set to the following values: $\lambda_p = \lambda_{rec} = 200, \lambda_R = 2$ and $\lambda_d = 5$.

6.1.2 Gaze estimation

The gaze estimation network E_{gaze} learns to predict the 3D direction of gaze from an input image of each eye. Particularly, our aim is to recover spherical coordinates (θ, ϕ) that represent gaze direction and employ them for training as well as driving our reenactment system. To that end, we adopt a mesh regression approach to gaze estimation, meaning that we train E_{gaze} to predict 3D meshes of eyes instead of 3D gaze vectors or angles. This approach is based on the fact that estimating dense geometry instead of few pose parameters, has recently been shown to benefit face and body pose estimation systems [DGV+20b, KGK+20, RAG18].

To train E_{gaze}, we first define a 3D eye mesh template, $\mathbf{T}_{eye}$, consisting of $N_v^{eye} = 481$ vertices and $N_t^{eye} = 928$ triangles, as shown in Fig. 6.2, and enforce our model's predictions to adhere to this template so that exact correspondence exists. As common gaze estimation datasets provide gaze annotations as 3D vectors, angles or points on screen [ZSFB15, KKK+16, KRS+19], we create training and validation data compatible with our mesh regression approach by automatically fitting $\mathbf{T}_{eye}$ on available images, based on 2D sparse landmarks around the iris contour and the available gaze labels. To obtain the iris landmarks we employ the network from [PZBH18], but any similar model could have been used.

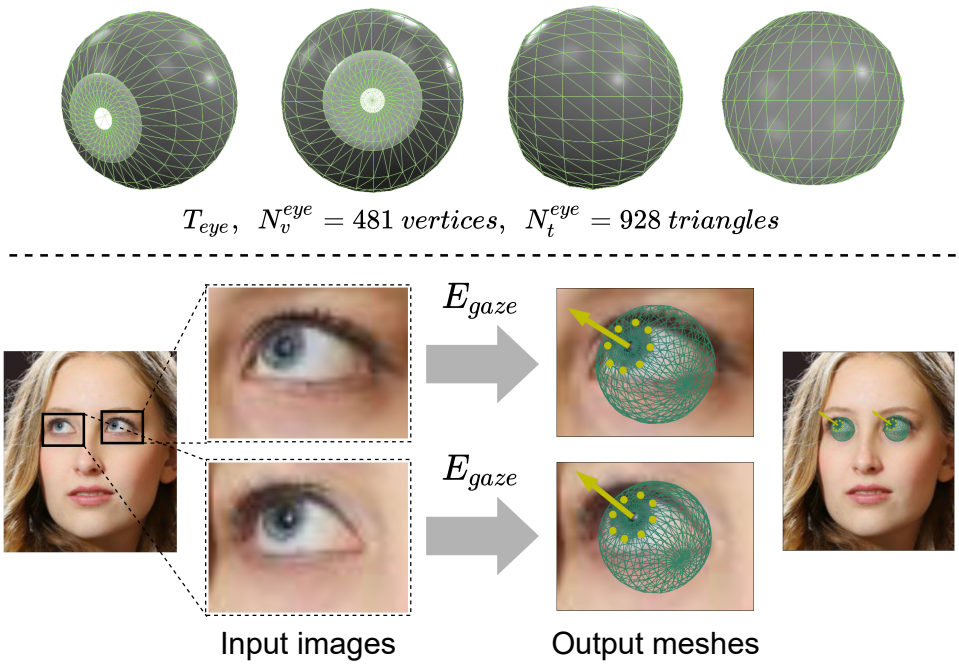


Figure 6.2: Outline of our gaze estimation method. Having defined an eyeball template, $\mathbf{T}_{eye}$ we generate pseudo-ground truth meshes for available gaze estimation datasets and employ them to train our 3D eye mesh regression network E_{gaze}, which given input images produces 3D coordinates adhering to $\mathbf{T}_{eye}$.

We adopt a simple architecture for E_{gaze}, consisting of a ResNet-34 backbone and a fully connected layer mapping eye image features to $\mathbf{v}$, which is a vector of $3 \times N_v^{eye}$ real values representing N_v^{eye} 3D eye coordinates on a normalized space. We optimize E_{gaze} in a supervised fashion, based on the 3D eye meshes we fitted on available gaze estimation datasets. To recover 3D points from images, we minimise the distance

$$\mathcal{L}^v = ||\mathbf{v} - \mathbf{v}^*||_1, \tag{6.7}$$

between predicted coordinates $\mathbf{v}$ and the corresponding pseudo-ground truth ones $\mathbf{v}^*$. Additionally, to maintain feasible eye shape, we minimize the distance

$$\mathcal{L}^e = ||\mathbf{e} - \mathbf{e}^*||_1, \tag{6.8}$$

between the $3N_t^{eye}$ edge lengths $\mathbf{e}$ computed from $\mathbf{v}$ and those, $\mathbf{e}^*$, computed from $\mathbf{v}^*$. We define edge length to be the euclidean distance between vertices of the the same triangle according to $\mathbf{T}_{eye}$. Lastly, to increase gaze estimation accuracy we employ the gaze loss

$$\mathcal{L}^g = (180/\pi) \arccos(\mathbf{g}^T \mathbf{g}^*), \tag{6.9}$$

where $\mathbf{g}$ and $\mathbf{g}^*$ are normalized 3D gaze vectors, calculated by the centers of the eye and iris, from the predicted and ground truth meshes respectively. Combining the above losses, the overall objective function for E_{gaze} is given as

$$\mathcal{L}_{E_{gaze}} = \lambda_v \mathcal{L}^v + \lambda_e \mathcal{L}^e + \lambda_g \mathcal{L}^g, \tag{6.10}$$

with the hyper-parameters set to the following values $\lambda_v = \lambda_e = 0.1$ and $\lambda_g = 1$. As in our

reenactment system we code gaze using spherical coordinates, we calculate the angles (θ, ϕ) from the predicted eye meshes using the gaze vector $\mathbf{g} = (g_x, g_y, g_z)$, as

$$\theta = \arctan \sqrt{g_x^2 + g_y^2}/g_z, \quad \phi = \arctan g_y/g_z. \tag{6.11}$$

6.1.3 Image Synthesis

We perform talking head synthesis with the assistance of an image-to-image translation network G, which is based on the generator of HeadGAN [DZS21], without AdaIN layers[HB17] as we do not process audio features. Given a source image $\mathbf{x}^s$ and a target frame $\mathbf{x}^t$, along with the corresponding facial key-points $\mathbf{p}^s$, $\mathbf{p}^t$ and gaze angles $\mathbf{g}^s$, $\mathbf{g}^t$, we draw a source and a target facial sketch (see Figure 6.3). Gaze angles are color coded in the sketches, within the pixels defined by key-points that belong to the eyes. These two sketches, together with the source image, serve as inputs to network G, which learns to generate a photo-realistic image $\tilde{\mathbf{x}}$ of the source, according to the target head pose and facial expression. Please note that in this chapter we refer to the reference image as the source (identity) and the driving image as the target (expression and pose). This is the opposite to the notation we followed in the previous chapters, nonetheless, we adopt it for consistency with the related literature. As in HeadGAN [DZS21], network G is comprised of two modules: a flow network and a rendering network.

First, the flow network extracts visual feature maps from the source image and its corresponding key-point sketch in multiple spatial resolutions: $\{\mathbf{h}^{(i)}\}_{i=1,\ldots,L}$, $L = 3$. Then, the target sketch is injected into the network through SPADE [PLWZ19] layers, as modulation input, and guides the prediction of the optical flow $\mathbf{w}$, which warps the source image to the target expression and pose. In addition, we perform warping of each feature map, in order to align them spatially with the desired expression and pose.

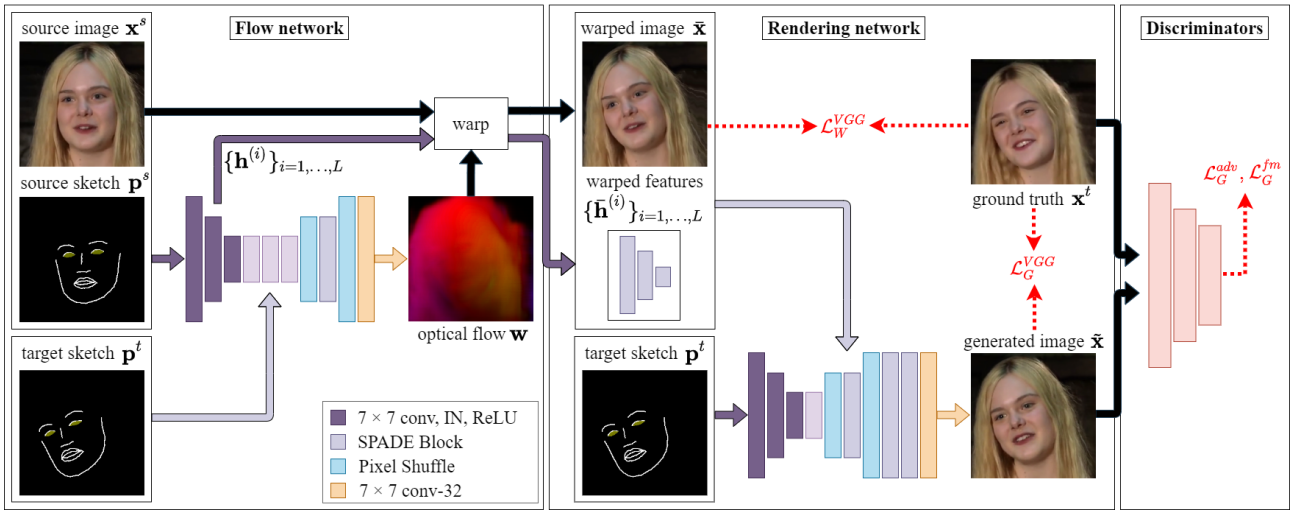


Figure 6.3: Overview of the image generation component. The generator is made up from two modules: a flow network and a rendering network. The flow network computes the optical flow for warping the reference image and features, according to the target sketch. The rendering network uses this visual information, in order to translate the driver's sketch into a photo-realistic image of the source. The two networks are trained jointly in an adversarial manner, with the assistance of image discriminators.

The rendering network first encodes the target sketch and then combines the resulting feature map with the warped features $\{\bar{\mathbf{h}}^{(i)} = \mathbf{w}(\mathbf{h}^{(i)})\}_{i=1,\dots,L}$ and warped image $\bar{\mathbf{x}} = \mathbf{w}(\mathbf{x}^s)$, which enter the network through SPADE [PLWZ19] layers, as modulation inputs. Up-sampling is performed with Pixel Shuffle [SCH+16] layers. The final output is a photo-realistic image $\tilde{\mathbf{x}}$ of the source identity, imitating the facial expressions and head pose shown in the target image.

N-shot extension We further extend our method to enable few-shot learning, in cases where more than one source images are available. For that, we propose an optional attention mechanism, with the addition of one more output layer to the flow network, which now learns to compute a set of 2D weights $\mathbf{m} \in \mathbb{R}^{H \times W}$, alongside the optical flow field. Given M source frames $\{\mathbf{x}^s_j\}_{j=1,\dots,M}$, we pass each one through the flow network, gaining flows $\{\mathbf{w}_j\}_{j=1,\dots,M}$ and weights $\{\mathbf{m}_j\}_{j=1,\dots,M}$. Next, the warped image is computed with the assistance of a Softmax function

$$\bar{\mathbf{x}} = \frac{\sum_j^M \exp(\mathbf{m}_j)\mathbf{w}_j(\mathbf{x}^s_j)}{\sum_j^M \exp(\mathbf{m}_j)} \tag{6.12}$$

and the warped features are given as

$$\bar{\mathbf{h}}^{(i)} = \frac{\sum_j^M \exp(\mathbf{m}_j)\mathbf{w}_j(\mathbf{h}^{(i)}_j)}{\sum_j^M \exp(\mathbf{m}_j)}, \quad i = 1, \dots, L. \tag{6.13}$$

Fig. 6.4 shows a visual example of flows and weights.

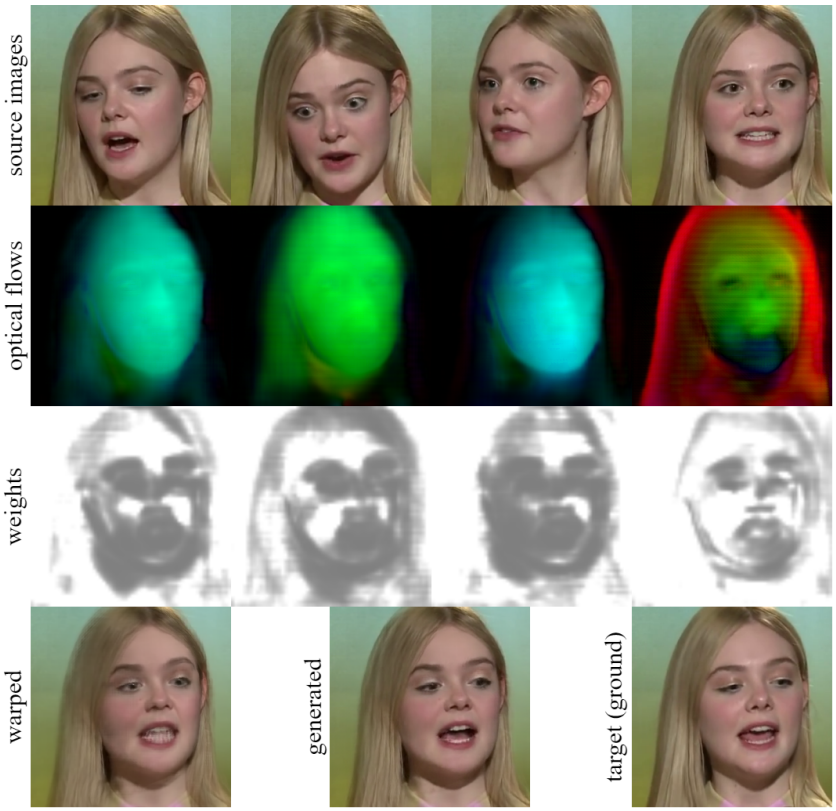


Figure 6.4: Illustration of optical flows and weights predicted by the flow network, in case four source images are provided instead of one (4-shot). The warped image is calculated from the source images, flows and weighs with the application of Eq. (6.12).

Training Objectives The flow network and rendering network that make G are trained jointly, on the task of self-reenactment (reconstruction). To that end, we sample the source image from the frames of the target video to obtain image pairs that belong to the same person and scene. In this case, the generated image $\tilde{\mathbf{x}}$ should match the target frame $\mathbf{x}^t$ that serves as ground truth. We optimise the generator G by minimising the distance between feature maps extracted from multiple layers of a pre-trained VGG network [SZ14]:

$$\mathcal{L}_G^{VGG} = \sum_l ||VGG_l(\tilde{\mathbf{x}}) - VGG_l(\mathbf{x}^t)||_1. \tag{6.14}$$

Additionally, we apply a VGG loss on the warped image, in order to force the flow network to learn a correct flow from the source image to the desired head pose, which gives the loss term $\mathcal{L}_F^{VGG}$. We further improve the photo-realism of synthetic images by placing an adversarial loss term $\mathcal{L}_G^{adv}$ on $\tilde{\mathbf{x}}$ and more specifically a Hinge GAN loss [LY17]. Similarly with [KDRZ20] and [DZS21], we employ two critics, a general image discriminator D_I and a dedicated discriminator D_M for the mouth area, which are optimised alongside G. Finally, we use these discriminators to compute visual features from both real (target) and synthetic images and compute a feature matching loss $\mathcal{L}_G^{fm}$ [WLZ+18b] that is known to be effective at increasing the photo-realism of synthetic samples. Summing up, the overall objective function for G is given by:

$$\mathcal{L}_G = \mathcal{L}_G^{adv} + \lambda_{VGG}(\mathcal{L}_G^{VGG} + \mathcal{L}_F^{VGG}) + \lambda_{FM}\mathcal{L}_G^{FM} \tag{6.15}$$

where $\lambda_{VGG} = \lambda_{FM} = 10$. Both discriminators are optimised under their corresponding adversarial loss terms and have a similar architecture with the one proposed in [PLWZ19].

6.1.4 Free-HeadGAN inference

For the task of reenactment during inference, the source and target images belong to different identities. In order to adapt the target key-points to the facial shape of the source identity, we use E_{can} to regress 3D key-points, head pose and expression deformations from $\mathbf{x}^s$, and then evaluate Eq. (6.1) to obtain the canonical key-points $\mathbf{p}^{s,can}$. At the same time, we estimate the target pose and expression $\mathcal{T}^t, \mathbf{d}^t$ from $\mathbf{x}^t$. The adapted target key-points $\mathbf{p}^{s,adapt}$ are obtained

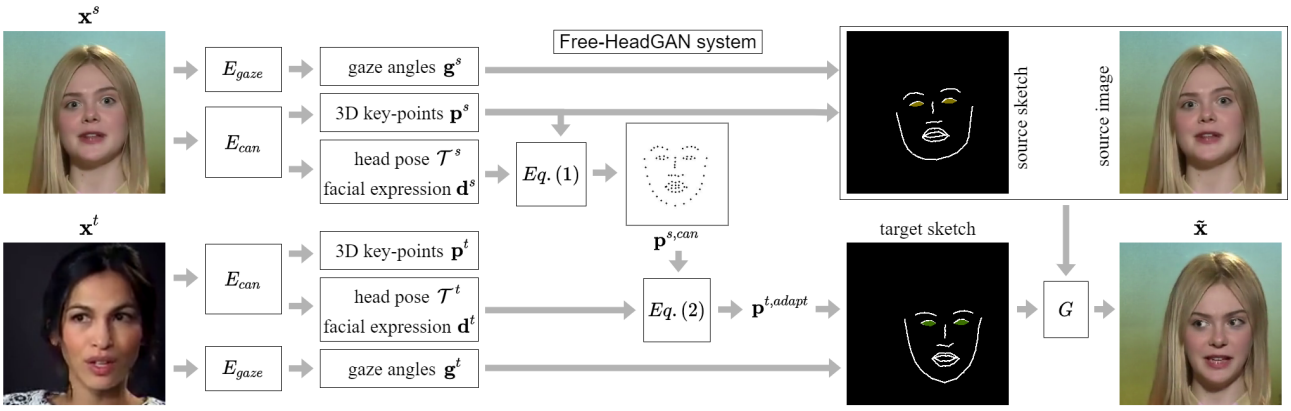


Figure 6.5: End-to-end pipeline of Free-HeadGAN during inference, for the task of reenactment. We adapt the target key-points to the facial shape (identity) of the source.

with Eq. (6.2), which transforms the source canonical key-points using the estimated target pose and expression. In this way, we remove the target identity-related information from key-points and inject the source one, which helps to overcome to problem of identity mismatching. In parallel, we estimate eye gaze from both images with network E_{gaze} and finally draw the sketches that serve as input to network G, which in turn creates the output image $\tilde{\mathbf{x}}$. For a visual inspection of our proposed pipeline during reenactment, please refer to Fig 6.5.

6.2 Experimental Results

6.2.1 Dataset and Training

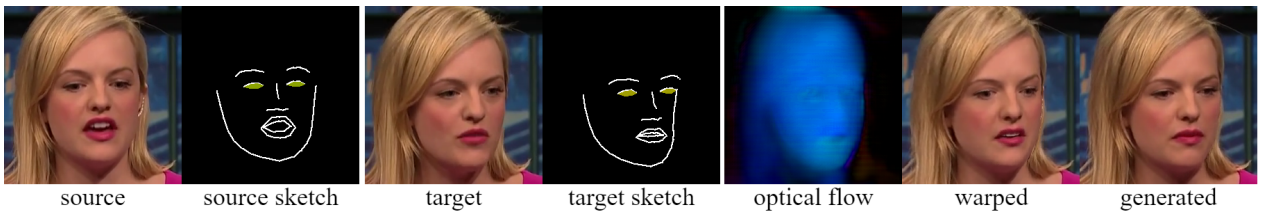
We train each component of Free-HeadGAN independently. Both the network that infers the canonical key-points E_{can} and the generative network G (with discriminators D_I, D_M) are optimised on VoxCeleb [NCZ17] video dataset, which contains over one hundred thousand clips and more than one thousand identities, at a 256×256 resolution. We keep the original train and test split. As a pre-processing step, we prepare three-dimensional 68 landmarks for each frame in the training split with RetinaFace model [DGV+20a], which serve as pseudo-ground truth annotations. We note that RetinaFace is pre-trained on WIDERFACE dataset [YLLT16]. For the extraction of ground truth head rotation angles, we employ a least-squares solver to determine the transformation between facial landmarks and a fixed landmark template that represents the frontal-neutral pose. The aforementioned annotations are used only for training Free-HeadGAN. During inference, our method requires only a pair of source and target images.

For the optimisation of network E_{can}, we use ADAM solver [KB14] with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and learning rate $\eta = 0.0002$. We use the same optimiser and hyper-parameters for training G and its adversaries D_I, D_M. All models are optimised for 5 epochs on the entire VoxCeleb database. We extend Free-HeadGAN to accommodate N-shot learning, by first training in one-shot without predicting attention weights. Then, we add the weight layer and fine-tune the GAN for one more epoch using two source images (2-shot), while freezing the parameters of the flow network.

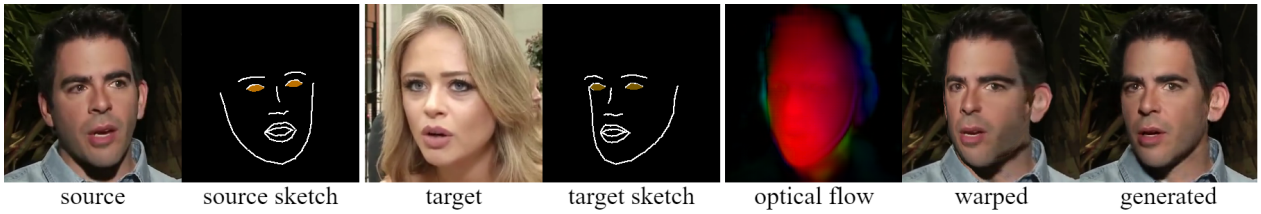
We optimize network E_{gaze} on a combination of recent gaze estimation datasets, aiming to include variation from multiple image domains. Particularly we employ ETH-XGaze [ZPB+20], which includes large variation in gaze and face pose and consists of 756 thousand frames from 80 subjects, Gaze360 [KRS+19] which is captured both indoors and outdoors and includes 127 thousand training sequences from 365 subjects and MPIIGaze [ZSFB15], which provides over 213 thousand frames of 15 subjects, captured with laptop cameras. We employ the default training and validation sets from Gaze360 and MPIIGaze, while we perform a manual split for ETH-XGaze, as gaze labels for its test data are not available. Lastly, we use ADAM [KB14] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning rate $\eta = 0.0001$.

6.2.2 Comparison with State-of-the-Art

We evaluate the reconstruction capabilities of our method quantitatively, using L1 distance (L1), Peak signal-to-noise ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS)



(a) Self-reenactment example



(b) Reenactment example

Figure 6.6: We evaluate the generative performance of Free-HeadGAN on the tasks of (a) same-identity reconstruction (self-reenactment) and (b) cross-identity motion transfer (reenactment).

[ZIE+18]. All three metrics measure the distance, between the synthesised and ground truth target frames. Furthermore, we assess the photo-realism of generated images with Fréchet Inception Distance (FID) [HRU+17, Sei20] and Fréchet Video Distance (FVD) [UvSK+18], as we handle video data and therefore crucial to measure the performance of models considering the temporal coherence among frames. The identity preservation in the synthesised data is calculated with Cosine Similarity (CSIM) between embedding vectors from the target and corresponding generated images. All embeddings are computed with the assistance of ArcFace [DGXZ19]. We use Action Units Hamming distance (AU-H) to measure expression transferability of models. OpenFace [ALS16] with [BMR15], runs on target and synthetic images for the detection of Action Units (AU). We measure the correctness of pose transfer with Average Rotation Distance (ARD), which is the $l1$-distance of Euler angles that correspond to head pose, between the target and generated frames in degrees. Finally, we evaluate gaze transfer with Average Gaze Distance (AGD). Given the gaze vector $\mathbf{g}^t$ regressed with E_{gaze} module from the target frame, as well as the vector $\tilde{\mathbf{g}}$ extracted from the corresponding synthetic image, we compute the angle between vectors as $(180/\pi)\arccos(\tilde{\mathbf{g}}^\top \mathbf{g}^t)$ and average across frames to obtain AGD in degrees.

We compare our one-shot Free-HeadGAN system both numerically and visually with state-of-the-art methods under two setups: a) same-identity reconstruction or self-reenactment, where the source and target identities coincide, and b) cross-identity motion transfer or reenactment, in which source and target identities are different. Please refer to Fig. 6.6 for a visualisation of the two tasks. More specifically, we compare our approach with X2Face [WKZ18], few-shot vid2vid (fs-vid2vid) [WLT+19], Bi-layer Neural Avatars (Bi-layer) [ZISL20], First Order Motion Model (FOMM) [SLT+19b] and HeadGAN [DZS21]. Please note that we tested two variations of FOMM, one with absolute (FOMM-abs) and one with relative key-point coordinates (FOMM-rel), which is the default setting for cross-identity motion transfer. For X2Face [WKZ18], FOMM [SLT+19b] and HeadGAN [DZS21], we use the models provided by their authors, all trained on VoxCeleb [NCZ17] dataset. For Bi-layer [ZISL20] method, we used the network parameters provided by its authors, trained on VoxCeleb2 dataset* [CNZ18]. We trained fs-vid2vid [WLT+19] using the official open source implementation, as pre-trained models are available. Given that the source code for MarioNETte [HKK+20], Warp-guided

GANs [GSZ+18] and face-vid2vid [WML21] is not publicly available, we were not able to measure Free-HeadGAN's performance against these systems.

In Table 6.2 and Table 6.1, we present a quantitative comparison of the proposed Free-HeadGAN with the aforementioned baselines. For self-reenactment, we reconstruct the entire test set of VoxCeleb, while we have chosen 15 pairs of reference images and driving videos for reenactment, the same as in [DZS21]. As suggested by the results, our method creates superior samples, with respect to image quality and reconstruction fidelity, both in same-identity reconstruction and cross-identity motion transfer. In addition, our approach preserves the identity of the source better in all experiments. For expression transferability, we observe that Free-HeadGAN is left slightly behind from [ZISL20] and [DZS21]. This is attributed to the following two reasons: First, the ground truth 3D facial landmarks annotations have been extracted with RetinaFace [DGV+20a], which is not a model focusing exclusively on landmark prediction and occasionally misses fine details. Second, during reenactment the adaptation of target key-points to the source identity with network E_{can} relies on the regression of expression deformations, which also comes with small inaccuracies. This is compromise we make, as we have chosen 3D facial key-points instead of 2D and we pay particular attention to the preservation of the source identity with network E_{can}.

Table 6.1: Numerical comparison with state-of-the-art methods on the task of self-reenactment (same-identity reconstruction) for VoxCeleb [NCZ17] test set.

	Self-reenactment					
Method	**L1** (↓)	**PSNR** (↑)	**LPIPS** (↓)	**FID** (↓)	**FVD** (↓)	**CSIM** (↑)
X2Face [WKZ18]	13.49	20.69	0.260	130.2	697	0.600
fs-vid2vid [WLT+19]	17.15	18.52	0.197	62.8	471	0.542
Bi-layer* [ZISL20]	12.18	20.19	0.152	92.2	394	0.590
FOMM-abs [SLT+19b]	12.34	20.93	0.153	64.9	338	0.754
HeadGAN [DZS21]	11.32	21.46	0.112	36.1	254	0.807
Free-HeadGAN	**9.96**	**22.16**	**0.100**	**35.4**	**248**	**0.810**

Table 6.2: Numerical comparison with state-of-the-art methods on the task of reenactment (cross-identity motion transfer) for VoxCeleb [NCZ17] test set.

	Reenactment				
Method	**FID** (↓)	**CSIM** (↑)	**ARD** (↓)	**AU-H** (↓)	**AGD** (↓)
X2Face [WKZ18]	122.1	0.520	4.39	0.346	21.4
Bi-layer* [ZISL20]	172.8	0.563	**1.01**	**0.296**	16.6
FOMM-abs [SLT+19b]	100.7	0.587	1.46	0.298	13.8
FOMM-rel [SLT+19b]	63.7	0.765	12.53	0.400	21.2
HeadGAN [DZS21]	58.0	0.688	1.35	0.326	16.8
Free-HeadGAN	**53.9**	**0.789**	1.26	0.351	**13.3**



Figure 6.7: Visual comparison of our method with baselines on the task of self-reenactment.



Figure 6.8: Qualitative evaluation of our method against baselines on the task of reenactment.

In Fig. 6.7 and Fig. 6.8 we show examples from a visual comparison of Free-HeadGAN with the baselines, on the task of reconstruction and reenactment respectively. As can be seen, the qualitative results confirm our numerical analysis. More specifically, our proposed system outperforms all state-of-the-art methods in terms of reconstruction fidelity, photo-realism, identity conservation and gaze transfer. The generated samples of Free-HeadGAN are as good as the ones of HeadGAN [DZS21], in terms of visual quality. Nonetheless, in comparison to [DZS21], Free-HeadGAN appears to be significantly more reliable when it comes to eye gaze transfer, thanks to our gaze prediction network and explicit gaze conditioning for image synthesis.

6.2.3 Ablation Study

We follow a few-shot learning approach to adjust our system in a setting where multiple source images are available. Although the extended flow network that predicts weights has been trained in a 2-shot scenario, during inference it can operate for any number of source frames N. We examine the setting where $N = 2, 4$ or 8 source frames are given and report the results on the task of reconstruction of VoxCeleb test set, in Table 6.3. As suggested by all evaluation metrics, the generative performance of our system improves when the number of available images increases. In Fig. 6.9 we present visual examples that show the effect of the number of source images N in the quality of synthesised data. The results clearly demonstrate the beneficial effect of N-shot learning.

Table 6.3: Evaluation of Free-HeadGAN N-shot extension, in case more than one source images are available.

N-shot	**L1** (↓)	**PSNR** (↑)	**LPIPS** (↓)	**FID** (↓)	**FVD** (↓)	**CSIM** (↑)
1-shot	9.96	22.16	0.100	35.4	248	0.810
2-shot	9.02	22.99	0.088	32.7	224	0.834
4-shot	7.90	24.01	0.077	30.7	207	0.857
8-shot	7.20	24.96	0.068	29.9	196	0.868

Next, we examine the significance of the two front-end components of our talking head synthesis system and more specifically networks E_{can} and E_{gaze}.

We focus on cross-identity motion transfer, as the canonical key-point estimator E_{can} is particularly developed to tackle the identity mismatches in reenactment, since it allows to adapt the target key-points to the facial shape of the source. In order to evaluate its contribution, we develop a variation of Free-HeadGAN by replacing E_{can} network with RetinaFace [DGV+20a]. That is, we utilise directly the 3D key-points regressed by RetinaFace in order to draw the sketches that serve as conditional input to the generator, without adapting them to the identity of the source. The quantitative results displayed in Table 6.4 indicate that removing E_{can} has a critical effect on CSIM metric that measures identity preservation. Moreover, the overall quality of samples degrades as FID score increases. On the contrary, expression transfer slightly improves, indicating that the canonical space key-points actually retain a small amount of expression information.

In order to validate the importance of explicit eye gaze conditioning in image synthesis, we test a second variation of our system, where we do not encode any gaze information into key-point



1-shot 2-shot 4-shot 8-shot

Figure 6.9: Synthetic examples when using one, two, four or eight source images in the N-shot scenario. The results improve in terms of image quality as we get access to more source inputs.

Table 6.4: Ablation study results on the significance of Free-HeadGAN components on the task of reenactment.

Variation	**FID** (↓)	**CSIM** (↑)	**AU-H** (↓)	**AGD** (↓)
Free-HeadGAN	**53.9**	**0.789**	0.351	**13.3**
Free-HeadGAN w/o E_{can}	66.6	0.640	0.304	-
Free-HeadGAN w/o E_{gaze}	-	-	-	17.7

sketches. To that end, we train a Free-HeadGAN generator that learns to transfer gaze direction relying solely on the facial key-points that belong to the eyes, as we do not color code the gaze angles inside the eye cavities. We evaluate the performance of this variation with AGD metric, which measures the average angular distance between the driving and generated gaze on the test set for reenactment. We display AGD results in Table 6.4, which confirms the significance of gaze estimator E_{gaze} quantitatively. Additionally, in Fig. 6.10 we illustrate some cases where the model variation without explicit gaze input fails to capture the direction of the target eyes.

Lastly, we quantitatively verify the benefit of implicitly inferring gaze from predicted dense 3D eye coordinates instead of learning to directly predict few gaze parameters. To that end, we train two versions of E_{gaze}, one predicting dense eye meshes as described in Section 6.1.2, E_{gaze}^{meshes}, and one predicting 3D gaze vectors, $E_{gaze}^{vectors}$. We perform within-dataset, cross-subject experiments on the recent gaze estimation datasets MPIIGaze [ZSFB15], Columbia [SYFN13], UTMV [SMS14], and Gaze360 [KRS+19]. We compute the AGD and present results in Table 6.5, which confirms our initial suggestion.



driver Free-HeadGAN Free-HeadGAN w/o E_{gaze}

Figure 6.10: Examples where explicit gaze control is essential for successful gaze transfer, as 68 facial landmarks appear to be insufficient.

Table 6.5: Comparison between the 3D mesh and 3D vector regression approaches to gaze estimation, on within-dataset, cross-subject experiments.

	AGD (↓)			
Variation	MPIIGaze	Columbia	UTMV	Gaze360
$E_{gaze}^{vectors}$	4.83	3.84	5.62	12.7
E_{gaze}^{meshes}	**3.35**	**1.12**	**4.11**	**10.4**

6.2.4 Pose and Gaze Manipulation

Apart from cross-identity motion transfer (reenactment) and video portrait compression and reconstruction (self-reenactment) we can use Free-HeadGAN to edit image portraits. That is, we can set the driving head pose manually, simply by rotating the 3D source key-points. Moreover, we can change the gazing direction of the reference subject, by editing the estimated gaze angles before feeding them to the generator through the target sketch. In Fig. 6.11 we demonstrate the ability of our proposed system to edit the head pose and eye gaze of any given reference image. Here, the left column shows the original reference image, while the next three columns depict synthesised images by our method, after editing pose and gaze. Our image editing task makes Free-HeadGAN a powerful tool for data augmentation, as our method can be used to synthesise a given image portrait in various poses. Considering that our system provides strong identity preservation, such image augmentation can benefit various computer vision tasks related to face recognition.

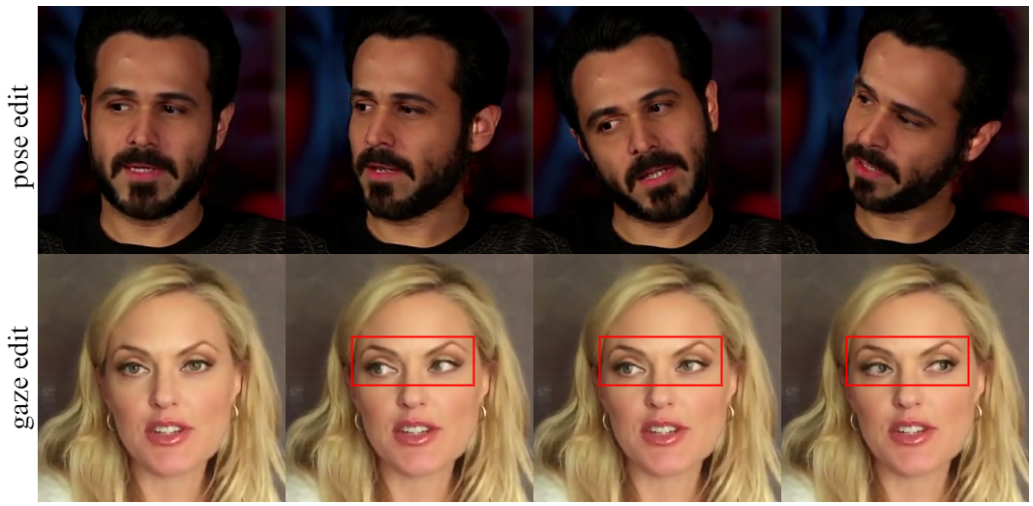


Figure 6.11: The 3D key-point modeling of faces enables our system to edit the head pose of a given reference image. Furthermore, we have explicit gaze control over the subject.

6.3 Limitations

While our method performs one-shot reenactment with unparalleled photo-realism and proves explicit gaze control and strong identity preservation properties, it does not come without any limitations. Considering that high quality data are crucial for training deep learning algorithms, the same applies to our system. In particular, Free-HeadGAN's performance in extreme target poses, such as large head rotations, highly depends on the quite limited pose distribution in open source databases such as VoxCeleb [NCZ17]. Moreover, as suggested by our qualitative analysis and FID scores [HRU+17], there exists a significant performance gap between self-reenactment and reenactment. Aside from the adaptation of key-points in cross-identity motion transfer, this gap is caused by the random selection of source and target image pairs during training and could be improved by a more sophisticated learning strategy, where progressively tougher pairs are chosen. Last but not least, even though our method does not rely on 3DMMs [BV99, PKA+09, BRP+18, CWZ+13, LBB+17], the supervision of our networks depends on the quality of 3D landmark pseudo-annotations with RetinaFace [DGV+20a] and labeled gaze data, which come with their own inaccuracies.

6.4 Summary

In this chapter, we presented Free-HeadGAN, a model that extends HeadGAN [DZS21] while releasing it from 3DMM fitting and 3D face rendering requirements during pre-processing. Instead, we condition synthesis on 3D landmarks which also support free-view synthesis. We employed a network that takes care of the identity mismatches in cross-identity motion transfer. Finally, we proposed a gaze estimation module and demonstrated the advantage on utilising gaze signals for precisely controlling the eye movements of generated samples. Please refer to Appendix B for more reenactment results.

Chapter 7

Conclusion

In this thesis, we proposed various methods which make progress towards generating photo-realistic video portraits of human faces in a controllable manner. In Chapter 3 and Chapter 4 we focused on person-specific approaches to the problem of full head reenactment. We trained identity-specific deep generative models that learn to transfer the facial expressions, head pose and eye gaze from any source actor to a target subject. The generated video samples are temporally coherent and appear convincing to human observers.

While the aforementioned methods are able to produce highly realistic video portraits, they require long training video data of the target subject and train a separate set of networks for each identity. In the second part of this thesis, we draw our attention to identity-agnostic models, which manage to synthesise convincing talking heads using only a single reference image of the target subject. This is a very challenging problem with inherent ambiguity on the appearance of faces. In Chapter 5 and Chapter 6, we present one-shot solutions for the task of reenactment, which demonstrate high levels of identity preservation and photo-realism.

In the remaining of this chapter, we summarise the various methods presented in this thesis, discuss their strengths and shortcomings and propose future directions to address their limitations.

7.1 Summary of Achievements

As opposed to face reenactment, full head reenactment methods aim to transfer the complete head pose and facial expression from a source video to a given target face. In Chapter 3, we presented a video-based solution to the problem. Inspired from recent advancements in the field of video-to-video translation, we devised Head2Head, a deep generative framework that relies on adversarial learning for reenacting human faces in a photo-realistic way. To that end, we adopted a sequential generator network that is trained jointly with a dynamics discriminator, which capitalises on an accurate optical flow of the face. The process of frame synthesis is conditioned on a semantic face representation, which enables to disentangle expression information from facial shape and helps to alleviate the identity gap when reenacting the target subject with a different source actor. Since this conditional representation does not provide any information inside the mouth cavity, we included a dedicated mouth discriminator to further enhance the

realism around the teeth and the interior of the generated mouth. We performed qualitative and quantitative experiments to demonstrate the merits of our method, along with a user study. Our study revealed that our reenactment system synthesises video samples indistinguishable from real ones. In many cases, our generated samples appear to be more convincing that video portraits produced by state-of-the-art reenactment methods. Finally, we demonstrated how our pipeline can be extended in the direction of multi-person synthesis, using concepts from few-shot learning.

The traditional architectures for person-specific reenactment, including that of Head2Head, rely on the encoder-decoder paradigm. That is, they receive a 2D input, such as a rendering of a 3D facial shape or a landmark-based representation (e.g. 2D sketch) and transform it to a photo-realistic frame. In Chapter 4, we proposed an alternative architecture for the problem of controllable video portrait synthesis, which consists of an 2D coordinated-based MLP coupled with a CNN decoder. This design choice removes the need to condition the generative process on face renderings created by physical-based renderers, as we feed the expression, pose and gaze parameters directly to the system thought an MLP network. Furthermore, this formulation enables our model to perform audio-driven talking head synthesis, a problem which has been studied separately from video-based reenactment in previous works. Our extensive experiments suggested that our novel architecture outperforms prior state-of-the-art methods on the tasks of self-reenactment, cross-identity reenactment and audio-driven video portrait synthesis. In addition, during inference, our generative model operates at real-time speeds, which are orders of magnitude faster than recent NeRF-like systems [GTZN21, GCL+21]. Unlike these NeRF-based systems, our method is not 3D aware. However, we have shown that for the purposes of video portrait synthesis this is not an essential requirement for achieving consistent results in multiple head poses.

Despite the remarkable advances in person-specific video portrait synthesis, these methods have their limitations. Perhaps their most significant limitation is that they require re-training a new set of neural networks for each target identity, using long training videos. In Chapter 5, we presented HeadGAN, a one-shot reenactment method that relies on a single image of the target identity. In contrast to previous attempts, our generative model depends on a 2D semantic facial representation, which is obtained using a very fast 3D face reconstruction algorithm. For that, we capitalised on the rich prior statistics of humans faces that are contained within 3DMMs. In this way, we effectively disentangled expression-related deformations from facial shape information, and manage to tackle the identity preservation problem during cross-identity reenactment. In order to obtain more accurate lips motions we further condition our neural rendering network on audio features. Our experimental evaluation demonstrated that our approach produces superior results in terms of photo-realism and identity preservation when compared to previous methods for one-shot talking head synthesis and motion transfer. Last but not least, we have shown that the adopted semantic face representation enables our system to perform expression and pose editing of image portraits, as well as face frontalisation.

In Chapter 6 we show Free-HeadGAN, a variation of HeadGAN that does not depend on 3D face reconstruction and 3DMMs. To that end, we replaced the semantic face representation input (PNCC) with image sketches of 3D key-points. In order to alleviate the identity preservation problem that arises with landmarks, we adopted a canonical 3D key-point estimator that regresses 3D pose and expression-related deformations. Moreover, apart from head pose and facial expressions, our reenactment method is capable of fully transferring the eye gaze from a driving actor to a source identity, as it offers explicit gaze control. For the accurate prediction

of gaze, we trained a neural network that estimates a dense 3D mesh of the eyes. We further experimented with an extension of our generator to accommodate few-shot learning using an attention mechanism, in case more than one reference images are available. Compared to HeadGAN, Free-HeadGAN achieves comparable photo-realism combined with superior identity preservation.

7.2 Ethical Considerations

Generative models that offer explicit control of the actions or motions of human faces could be potentially used for unethical purposes. For instance, they could be employed to synthesise videos of well-known individuals acting or talking in a provocative way. We would like to clarify that the intention of this thesis is to make advancements towards photo-realistic face synthesis and reenactment for beneficent applications. We do not condone using generative algorithms of human faces to spread fake news or deceive the public.

7.3 Future Directions

There are several directions one could pursue in a follow up work.

When it comes to person-specific models, training video data are captured using stationary cameras. An interesting direction would be to model the 3D head with a implicit NeRF-like approach while considering videos portraits captured by a moving camera from multiple viewpoints. This formulation would enable the development of a 3D aware method that combines explicit head pose and expression control along with multi-view 3D consistency of the entire scene.

Moreover, one could overcome the person-specific limitations of Dynamic Neural Portraits (Chapter 4), by extending its architecture to accommodate few-shot or one-shot learning. Instead of training a new model from scratch for each target identity, we could utilise a mupliperson video database and design an identity-agnostic method. This could be achieved with the assistance of adaptive normalisation layers.

One of the main limitations of current reenactment methods is related to expression tracking. Given that most systems rely either on 3DMMs or sparse landmark detectors, the quality of expression transfer is restricted by the variability of expressions within 3DMMs or the limited expressiveness of few facial landmarks. Therefore, as a next step, one could concentrate on extracting detailed facial expressions that include as less identity information as possible. Apart from utilising existing emotion recognition networks, a promising direction would be to focus on learning expression embeddings from video data in an unsupervised way, using analysis-by-synthesis approaches.

Appendix A

HeadGAN Additional Results

In Fig. A.1 and Fig. A.2 we present more generated samples with HeadGAN (Chapter 5) using VoxCeleb test set [NCZ17]. Here, we also illustrate the reference and driving conditional PNCC inputs, as well as the predicted optical flow and warped image. To be more specific, in Fig. A.1 we show frames generated with same-identity reenactment or reconstruction. In Fig. A.2 we demonstrate the generative ability of HeadGAN for cross-identity reenactment.



Figure A.1: Reconstruction examples with HeadGAN. From left to right: reference, reference PNCC image, driving PNCC image, flow, warped, generated, driving.



Figure A.2: Reenactment examples with HeadGAN. From left to right: reference, reference PNCC image, driving PNCC image, flow, warped, generated, driving.

Appendix B

Free-HeadGAN Additional Results

In Fig. B.1 to B.6 we illustrate more samples synthesised with Free-HeadGAN model (Chapter 6) using VoxCeleb test set [NCZ17], for the task of cross-identity reenactment.

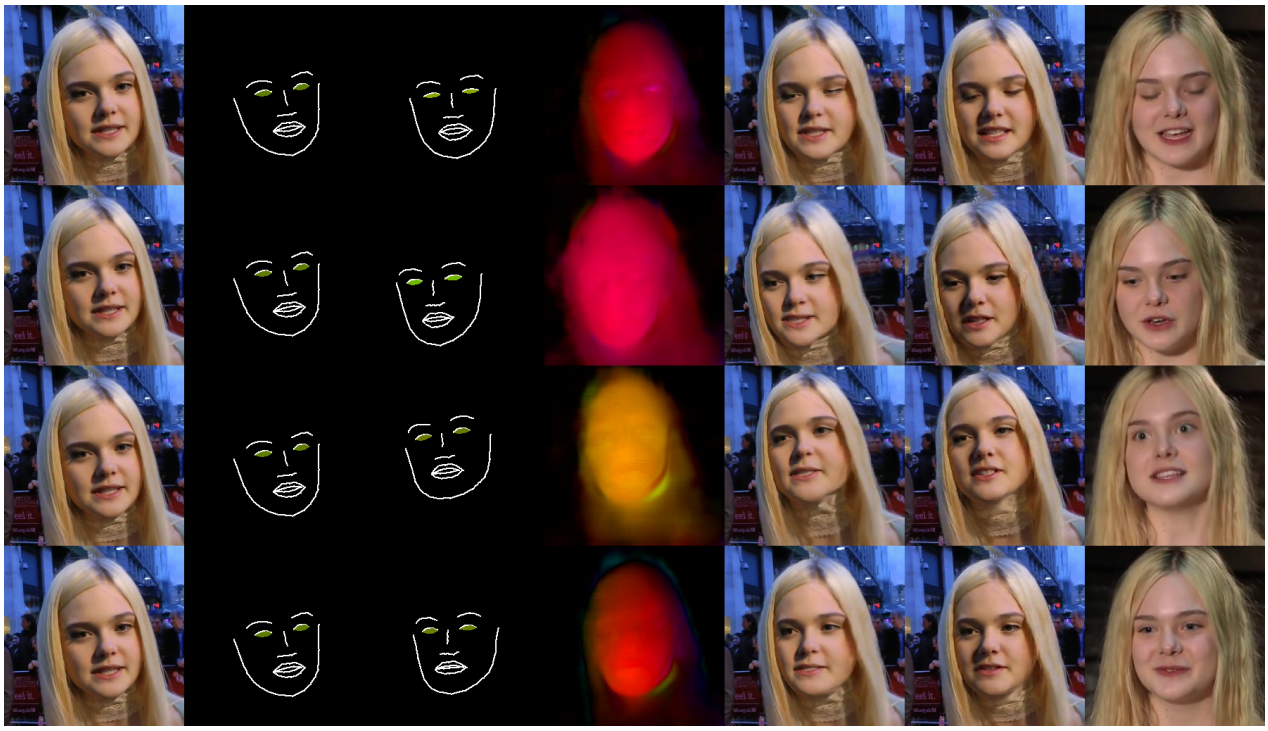
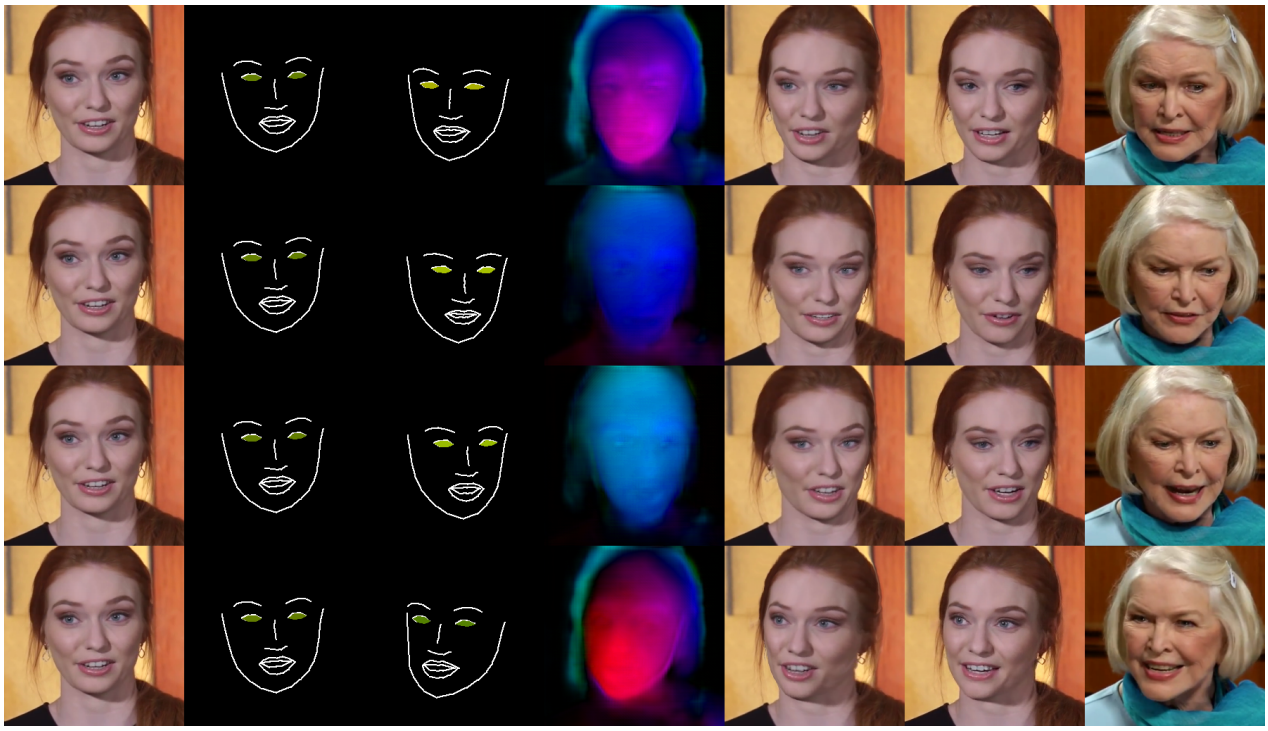
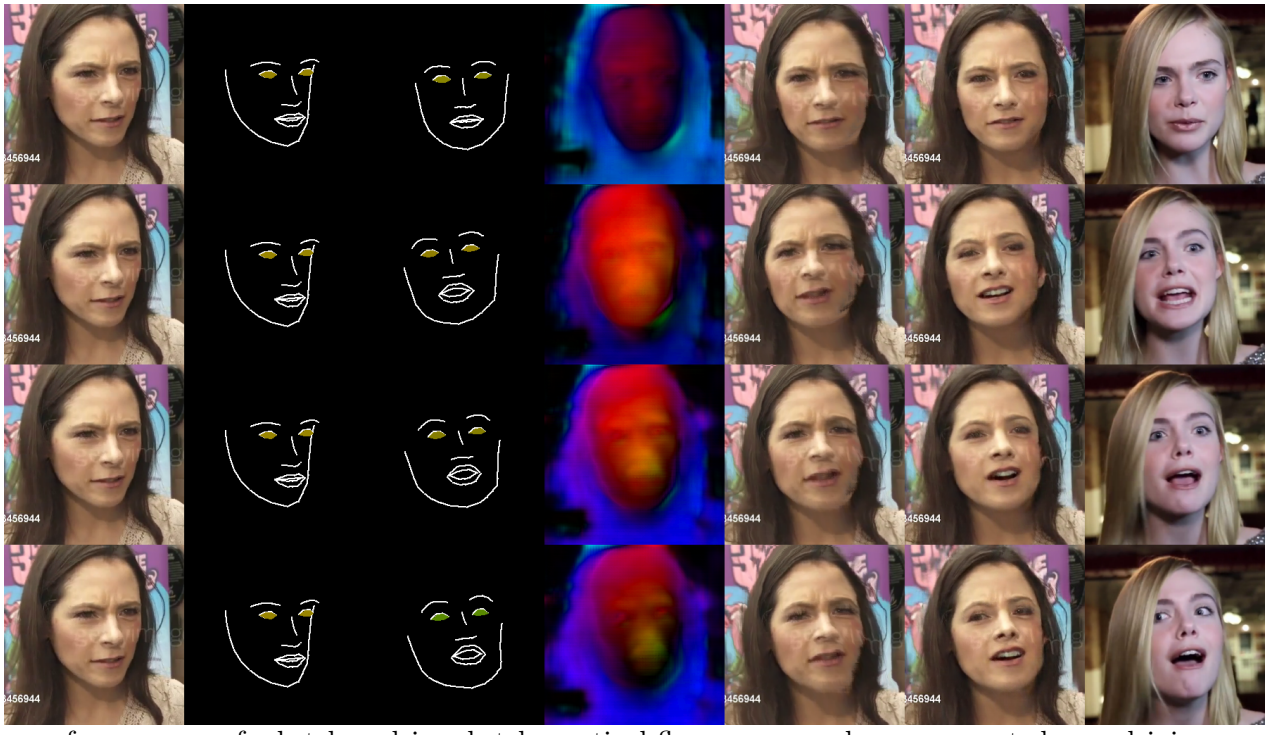


Figure B.1: Reenactment results with Free-HeadGAN (a)



reference ref. sketch driv. sketch optical flow warped generated driving

Figure B.2: Reenactment results with Free-HeadGAN (b)



reference ref. sketch driv. sketch optical flow warped generated driving

Figure B.3: Reenactment results with Free-HeadGAN (c)

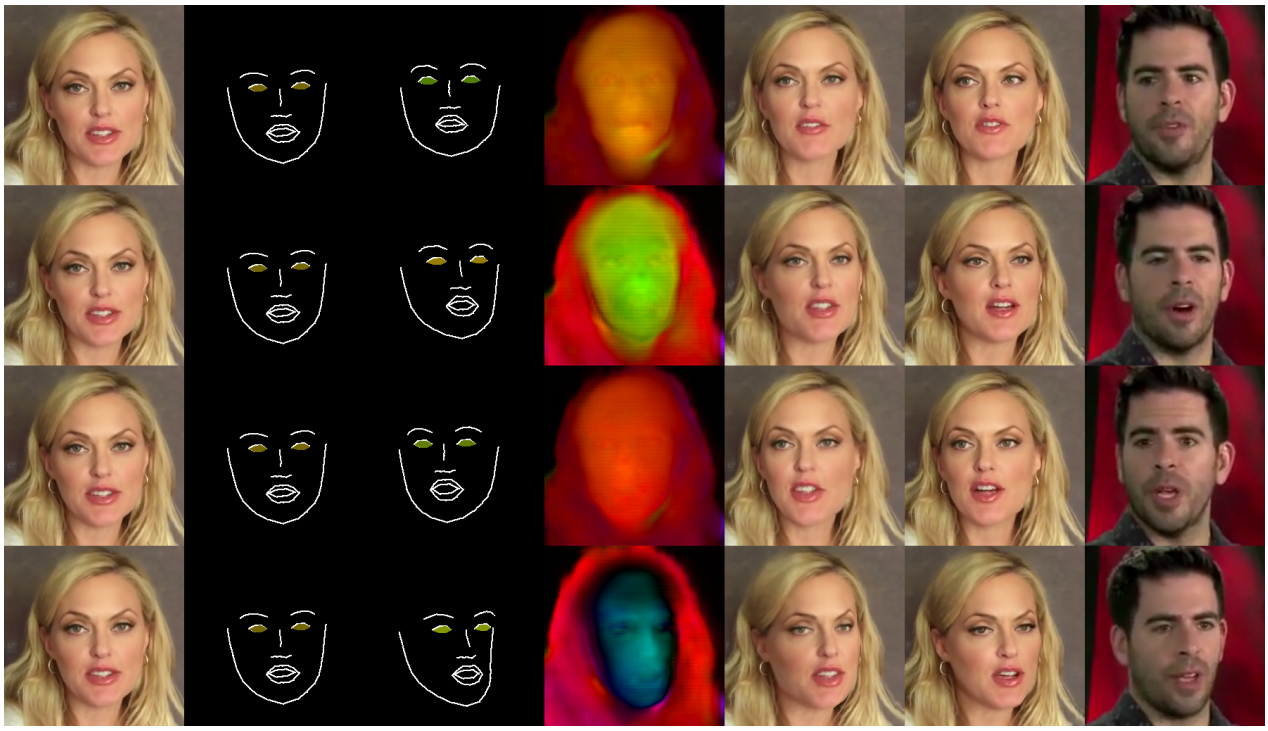


Figure B.4: Reenactment results with Free-HeadGAN (d)

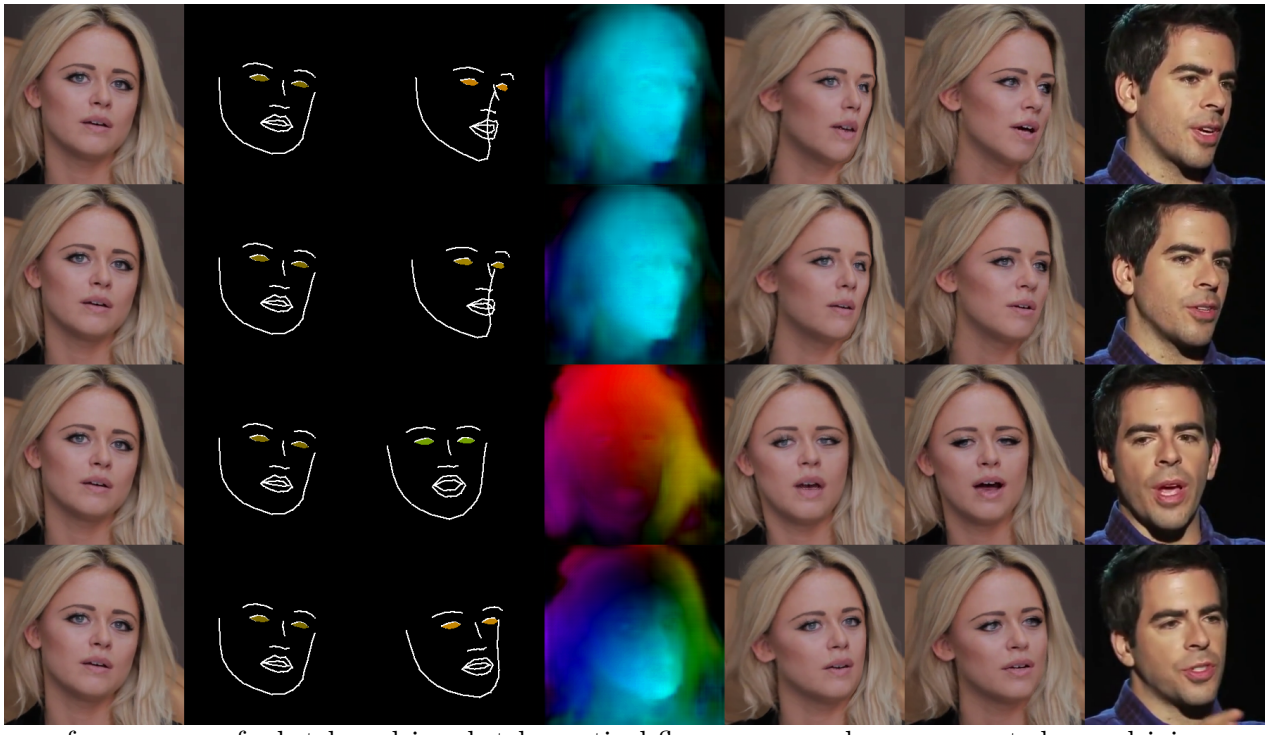


Figure B.5: Reenactment results with Free-HeadGAN (e)



Figure B.6: Reenactment results with Free-HeadGAN (f)

Bibliography

[AAA+16] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016.

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[ADK+21] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021.

[AECOKC17] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F Cohen. Bringing portraits to life. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.

[ALS16] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[AXS+22] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20364–20373, 2022.

[BAP+17] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3d face morphable models" in-the-wild". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 48–57, 2017.

[BAPD13] Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Proceedings of the IEEE international conference on computer vision*, pages 1513–1520, 2013.

[BMR15] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–6. IEEE, 2015.

[BRP+18] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *International Journal of Computer Vision*, 126(2):233–254, 2018.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[CCK+18] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[CKPZ18] Shiyang Cheng, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. 4dfab: A large scale 4d database for facial expression analysis and biometric applications. In *CVPR*, 2018.

[CL96] Thomas F Coleman and Yuying Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 1996.

[CMZP17] Shiyang Cheng, Ioannis Marras, Stefanos Zafeiriou, and Maja Pantic. Statistical non-rigid icp algorithm and its application to 3d face alignment. *Image and Vision Computing*, 58:3–12, 2017.

[CNZ18] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018.

[CS18] Zhaokang Chen and Bertram E Shi. Appearance-based gaze estimation using dilated-convolutions. In *Asian Conference on Computer Vision*, pages 309–324. Springer, 2018.

[CUYH20] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[CWZ+13] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2013.

[CZ16] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Asian conference on computer vision*, pages 87–103. Springer, 2016.

[DB16] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[DGV+20a] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[DGV+20b] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020.

[DGXZ19] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[DKS+21] Michail Christos Doukas, Mohammad Rami Koujan, Viktoriia Sharmanska, Anastasios Roussos, and Stefanos Zafeiriou. Head2head++: Deep facial attributes re-targeting. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(1):31–43, 2021.

[DR21] Adrienne De Ruiter. The distinct wrong of deepfakes. *Philosophy & Technology*, 34(4):1311–1332, 2021.

[DRC+19] Jiankang Deng, Anastasios Roussos, Grigorios Chrysos, Evangelos Ververas, Irene Kotsia, Jie Shen, and Stefanos Zafeiriou. The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking. *International Journal of Computer Vision*, 127(6):599–624, 2019.

[DTSB15] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1538–1546, 2015.

[DZS21] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14398–14407, 2021.

[EST+20] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020.

[FFBB21] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. volume 40, 2021.

[FKA+18] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2235–2245, 2018.

[GAA+17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[GCL+21] Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5784–5794, 2021.

[GEB16] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[Gia15] Theodoros Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one*, 10(12):e0144610, 2015.

[GLWT21] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[GPKZ19] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[GSZ+18] Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. Warp-guided gans for single-photo facial animation. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018.

[GTZN21] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021.

[GVR+14] Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormahlen, Patrick Perez, and Christian Theobalt. Automatic face reenactment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4217–4224, 2014.

[GVS+15] Pablo Garrido, Levi Valgaerts, Hamid Sarmadi, Ingmar Steiner, Kiran Varanasi, Patrick Perez, and Christian Theobalt. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Computer graphics forum*, volume 34, pages 193–204. Wiley Online Library, 2015.

[GYZ+20] Zidong Guo, Zejian Yuan, Chong Zhang, Wanchao Chi, Yonggen Ling, and Shenghao Zhang. Domain adaptation gaze estimation by embedding with prediction consistency. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[HB17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[HCC+14] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[HKK+20] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10893–10900, 2020.

[HPV+19] Junfeng He, Khoi Pham, Nachiappan Valliappan, Pingmei Xu, Chase Roberts, Dmitry Lagun, and Vidhya Navalpakkam. On-device few-shot personalization for real-time gaze estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[HRU+17] 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. *Advances in neural information processing systems*, 30, 2017.

[IMS+17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[JAFF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[KALL17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KDMI+21] Rakshit Kothari, Shalini De Mello, Umar Iqbal, Wonmin Byeon, Seonwook Park, and Jan Kautz. Weakly-supervised physically unconstrained gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9980–9989, June 2021.

[KDRZ20] Mohammad Rami Koujan, Michail Christos Doukas, Anastasios Roussos, and Stefanos Zafeiriou. Head2head: Video-based neural head synthesis. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 16–23. IEEE, 2020.

[KGK+20] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020.

[KGT+18] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018.

[Kin09] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[KKK+16] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[KLA+20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[KRS+19] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6912–6921, 2019.

[KRZ20] Mohammad Rami Koujan, Anastasios Roussos, and Stefanos Zafeiriou. Deepfaceflow: in-the-wild dense 3d facial motion estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6618–6627, 2020.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[KVH84] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[KWRB11] Martin Koestinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 2144–2151. IEEE, 2011.

[LBB+17] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017.

[Lev90] Marc Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics (TOG)*, 9(3):245–261, 1990.

[LLWL20] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5548–5557, 2020.

[LLWL21] Yunfei Liu, Ruicong Liu, Haofei Wang, and Feng Lu. Generalizing gaze estimation with outlier-guided collaborative adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3835–3844, 2021.

[LQG19] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 529–537, 2019.

[LY17] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[Mar19] Marek. Faceswap. github.com/MarekKowalski/FaceSwap, 09/2019.

[Max95] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[MBRS+21] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.

[MKKY18] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[MLX+17] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[MO14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv: 1411.1784*, 2014.

[MST+20] 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 *European conference on computer vision*, pages 405–421. Springer, 2020.

[NCZ17] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[NG21] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[PCPMMN21] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[PKA+09] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009.

[PLWZ19] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

[PMM+19] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9368–9377, 2019.

[PSB+21] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[PSH18] Seonwook Park, Adrian Spurr, and Otmar Hilliges. Deep pictorial gaze estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 721–738, 2018.

[PSH+21] Keunhong Park, Utkarsh 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. *arXiv preprint arXiv:2106.13228*, 2021.

[PZBH18] Seonwook Park, Xucong Zhang, Andreas Bulling, and Otmar Hilliges. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In *Proceedings of the 2018 ACM symposium on eye tracking research & applications*, pages 1–10, 2018.

[RAG18] Iasonas Kokkinos Riza Alp Guler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018.

[RAP+21] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[RCV+19] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[SBFB19] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7763–7772, June 2019.

[SCH+16] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[Sei20] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.1.1.

[SKCK17] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: A pixelcnn implementation with discretized logistic mixture likelihood and other modifications. In *ICLR*, 2017.

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[SLT+19a] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019.

[SLT+19b] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 32, 2019.

[SMS14] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1821–1828, 2014.

[SSKS17] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[ST17] Themos Stafylakis and Georgios Tzimiropoulos. Combining residual networks with lstms for lipreading. *arXiv preprint arXiv:1703.04105*, 2017.

[STZP13] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge.

In *Proceedings of the IEEE international conference on computer vision workshops*, pages 397–403, 2013.

[SYFN13] Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 271–280, 2013.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[SZSC21] Yunjia Sun, Jiabei Zeng, Shiguang Shan, and Xilin Chen. Cross-encoder for unsupervised gaze representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3702–3711, October 2021.

[TET+20] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. *ECCV 2020*, 2020.

[tor19] torzdf. Deepfakes. github.com/deepfakes/faceswap, 09/2019.

[TTG+21] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021.

[TZN+15] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183–1, 2015.

[TZN19] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM TOG 2019*, 2019.

[TZS+16] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[TZT+18] Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner. Headon: Real-time reenactment of human portrait videos. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018.

[UVL16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[UvSK+18] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[VdOKE+16] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[VOKK16] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[WKZ18] Olivia Wiles, A Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–686, 2018.

[WLT+19] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. *arXiv preprint arXiv:1910.12713*, 2019.

[WLZ+18a] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018.

[WLZ+18b] 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*, pages 8798–8807, 2018.

[WML21] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10039–10049, 2021.

[WQY+18] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2138, 2018.

[WZJ18] Kang Wang, Rui Zhao, and Qiang Ji. A hierarchical generative model for eye image synthesis and eye gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[YLLT16] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016.

[YLO19] Yu Yu, Gang Liu, and Jean-Marc Odobez. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11937–11946, 2019.

[YO20] Yu Yu and Jean-Marc Odobez. Unsupervised representation learning for gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[ZCR+17] Stefanos Zafeiriou, Grigorios G Chrysos, Anastasios Roussos, Evangelos Ververas, Jiankang Deng, and George Trigeorgis. The 3d menpo facial landmark tracking challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2503–2511, 2017.

[ZIE+18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[ZISL20] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. Fast bi-layer neural synthesis of one-shot realistic head avatars. In *European Conference on Computer Vision*, pages 524–540. Springer, 2020.

[ZLL+16] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155, 2016.

[ZLL+20] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5911–5920, 2020.

[ZPB+20] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020.

[ZSBL19] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9459–9468, 2019.

[ZSFB15] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520, 2015.

[ZXNT21] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. Cips-3d: A 3d-aware generator of gans based on conditionally-independent pixel synthesis. *arXiv preprint arXiv:2110.09788*, 2021.