Video Synthesis of a Talking Head

Mohammed Mesfer A Alghamdi

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

The University of Leeds

Faculty of Engineering and Physical Sciences

School of Computing

July 2023

Abstract

The ability to synthesise a video talking head from speech audio has many potential applications, such as video conferencing, video animation production and virtual assistants. Although there has been considerable prior work on this task, the quality of generated videos is typically limited in terms of overall realism and resolution. In this thesis, we propose a novel approach for synthesis of a high-resolution talking head video (1024x1024 in our experiments) from speech audio and a single identity image.

The approach is built on top of a pre-trained StyleGAN image generator. We model trajectories in the latent space of the generator conditioned on speech utterances. To train this model, we use a dataset of talking-head videos, which are mapped into the latent space of the image generator using an image encoder that is also pre-trained. We train a recurrent neural network to map from speech utterances to sequences of displacements in the latent space of the image generator. These displacements are applied to the back-projection into the latent space of a single identity frame chosen from a target video in the training dataset.

The thesis begins by reporting on an experimental evaluation of existing GAN inversion methods that map video frames into the latent space of a pre-trained StyleGAN image generator. We apply one such inversion method to train an unconditional video generator that requires an identity image and a random seed for the dynamical process that generates a trajectory through the latent space of the image generator.

We evaluate our method for talking head synthesis from speech audio with standard measures and show that it significantly outperforms recent state-of-the-art methods on commonly used audio-visual talking-head datasets (GRID and TCD-TIMIT). We perform the evaluation with two versions of StyleGAN; one trained on video frames depicting talking heads and the other on faces with static expressions (i.e., not talking). The quality of the results is shown to be better

when using StyleGAN pre-trained on talking heads. However, the range of possible identities is narrower due to the much smaller set of identities in the talking head dataset. The videos from experiments can be found at https://mohammedalghamdi.github.io/phd-thesis-website/

Declaration

The candidate confirms that the work submitted is his/her own, except where work which has formed part of a jointly authored publication has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some parts of the work presented in Chapters 4 and 5 have been published in the following:

- Mohammed M. Alghamdi, He Wang, Andrew J. Bulpitt, and David C. Hogg. **"Talking Head from Speech Audio using a Pre-trained Image Generator"** In Proceedings of the 30th ACM International Conference on Multimedia, 2022.
- Mohammed M. Alghamdi, He Wang, Andrew J. Bulpitt, and David C. Hogg. **"Video Synthesis of a Talking Head"** The British Machine Vision Association Symposium, 2022.

The above publications are primarily the work of the candidate.

Acknowledgements

I am very grateful to Professor David Hogg, my supervisor, for giving me the opportunity to be his PhD student. His profound knowledge, critical thinking, and constructive feedback have always encouraged me to become a better researcher. I would also like to express my gratitude to Professor Andy Bulpitt, my secondary supervisor, for his support and insightful input throughout my PhD journey. Working under their guidance has been a truly fortunate experience.

I extend my thanks to Dr. He Wang, my colleague, for his valuable input and helpful comments during my PhD studies. I would also like to acknowledge and thank my lab mates, Rebecca Stone and Jose Martinez, for their engaging discussions and support during this time. Furthermore, I would like to express my appreciation to other lab members Fangjum Li, Abrar Mohammed, Saud Althabiti, Abdullah Alsaleh, and Maan Alhazmi for their contributions.

I want to express my deepest gratitude to my parents, Mesfer and Azah, for their unconditional love. To my wife Nadia, I am very thankful for her constant support and understanding. My son, Zain, has brought endless joy to my life. I am also grateful to my siblings, Hanan, Abdullah, Abeer, Nada, and Asma, for their support.

Additionally, I would like to sincerely appreciate Taif University for granting me an exceptional scholarship to pursue my master's and PhD studies. The computational resources provided by the HPC members have been indispensable in the successful completion of my PhD.

Overall, I am profoundly grateful to all the teachers who have played a significant role in my lifelong learning journey, shaping me into the person I am today.

Contents

1 Introduction **4**

1.1 Introduction 4

1.2 Applications 7

1.3 Research motivation 7

1.4 Contributions 9

1.5 Publications 9

1.6 Thesis outline 10

2 Background **11**

2.1 Audio-driven talking-head videos 11

2.1.1 Learning-based methods 12

2.1.2 Sample-based methods 12

2.1.3 Deep learning methods 13

2.2 Talking-head datasets 17

2.2.1 Lab-based datasets 17

2.2.2 Datasets from the wild 17

2.3 Generative Adversarial Network (GAN) 18

2.4 Image synthesis using GANs 21

2.4.1 Style-based Generator: StyleGAN 23

2.5 GAN inversion 23

2.5.1 Encoder-based GAN inversion 26

2.5.2 Optimization-based GAN inversion 27

2.5.3 Hybrid-based GAN inversion 27

2.5.4 Pros and Cons of GAN Inversion Methods 29

2.6 Latent space understanding 29
2.7 Unconditional video generation 31
2.8 Conclusion 32

3 Video generation 35

3.1 Introduction 35
3.2 Datasets 36
3.3 GAN inversion for video projection 37
3.3.1 Optimisation-based GAN inversion 37
3.3.2 Encoder-based GAN inversion 48
3.4 Unconditional video generation 55
3.4.1 The approach 56
3.4.2 Framework 56
3.4.3 Training 57
3.4.4 Implementation 58
3.4.5 Results 59
3.4.6 StyleGAN image generator pre-trained on FFHQ dataset 59
3.5 Conclusion 66

4 Audio-driven Video Generation 70

4.1 Introduction 70
4.2 First model 71
4.2.1 Framework 71
4.2.2 Training strategy 74
4.2.3 Dataset 76
4.2.4 Implementation 77
4.2.5 Results 77
4.3 Second model 81
4.3.1 Framework 82
4.3.2 Training strategy 84
4.3.3 Dataset 86
4.3.4 Implementation 87
4.3.5 Results 88

4.4 Conclusion . . . 88

5 Evaluation **93**

5.1 Introduction . . . 93

5.2 Quantitative evaluation . . . 94

5.2.1 Measures . . . 94

5.2.2 Results . . . 97

5.3 Qualitative evaluation . . . 98

5.4 User study . . . 102

5.5 Conclusion and discussion . . . 104

6 Conclusion **106**

6.1 Limitations . . . 107

6.2 Ethical consideration . . . 108

6.3 Future work . . . 109

A Supplementary results for Chapter 3 **112**

A.1 Optimisation-based GAN inversion . . . 112

References **115**

List of Figures

1.1 The task of audio-driven talking-head video generation. The generative model takes an identity image and speech audio as inputs and generates a video of the identity, lip-synced with the given speech audio. 5

1.2 An overview of our novel approach to synthesise talking-head videos from an identity image and speech audio. 8

2.1 Figure shows generated videos using five different methods [Chung et al. 2017a; Vougioukas et al. 2018b; Chen et al. 2019b; Y. Zhou et al. 2020a; K R Prajwal et al. 2020a] 15

2.2 Samples selected from lab-based datasets 18

2.3 Samples selected from datasets from the wild 19

2.4 The GAN framework, figure taken from Goodfellow 2016 20

2.5 The figure shows images generated using the original GAN model and the DC-GAN model 22

2.6 The StyleGAN generator architecture, figure adapted from Karras et al. 2019. Here "A" stands for a learned affine transformation. 24

2.7 The figure shows images generated using the ProgressiveGAN model and the StyleGAN model 25

2.8 Qualitative results of different GAN inversion methods. The first column represents the input image and the second column shows the output of the optimisation method introduced by Abdal et al. 2019b. The remaining two columns show the results of using encoder-based methods: pSp encoder [Richardson et al. 2021] and ReStyle encoder [Alaluf et al. 2021], respectively. Figure is adapted from Alaluf et al. 2021 28

2.9 The figure shows the results of various methods examining the latent space of the StyleGAN for image manipulation. 30
2.10 The figure shows video samples generated using three different video generative models built on top of the StyleGAN model. 33

3.1 Three latent spaces of the StyleGAN model [Karras et al. 2019.] 38
3.2 Sample images from the FFHQ dataset used for training StyleGAN model [Karras et al. 2019]. 39
3.3 Sample images generated using the StyleGAN model trained on the FFHQ dataset [Karras et al. 2019]. 40
3.4 Overview of pre-processing a video frame from the TCD-TIMIT dataset. We employ the same methods used to pre-process the FFHQ dataset for training the StyleGAN model. 40
3.5 A comparison between a video projection using the optimization-based method into the W latent space (middle row) and the W^+ latent space (bottom row). The real video (first row) is from the TCD-TIMIT dataset. 42
3.6 Comparison of the loss values during the optimization process when projecting a video frame into W and W+ spaces using an optimisation-based approach. The video frame is taken from the TCD-TIMIT dataset. 43
3.7 Visualisation of two distinct projection attempts of the same video using the first two principal components computed per video. It is evident that the video projection using Algorithm 1 of the optimization-based method resulted in two different projections. The video sample comes from the TCD-TIMIT dataset. . 43
3.8 Visualisation of the video projection results using Algorithm 1 and Algorithm 2, based on the first two principal components computed for each video. Algorithm 2 projects a video as a smooth trajectory within the latent space. The video sample is taken from the TCD-TIMIT dataset. 44
3.9 The middle row shows video frames projected into the W^+ latent space with the use of Algorithm 1, while the bottom row displays the results of using Algorithm 2. The original video frames, shown in the first row, come from the TCD-TIMIT dataset. Visually, it can be seen that the quality of the video frames projected using Algorithm 2 is comparable to those produced by Algorithm 1. 45

3.10 Sample images of speakers from the TCD-TIMIT talking-head video dataset [Harte and Gillen 2015]. This dataset is used to train the StyleGAN model. . . . 46

3.11 Image samples generated using the StyleGAN generator pre-trained on video frames from the TCD-TIMIT talking-head dataset. 47

3.12 Each row displays the results of interpolating between two latent vectors in the latent space of the StyleGAN model, pre-trained on the TCD-TIMIT dataset. The first two rows demonstrate the results of interpolation performed in the Z space, while the latter two rows present the results of interpolation in the W space. *lerp* denotes linear interpolation, whereas *slerp* denotes spherical interpolation. It is clear that interpolation in the W space yields more visually appealing video frames, as highlighted in yellow. 48

3.13 The results of projecting video frames into the latent space of StyleGAN, pre-trained on the TCD-TIMIT dataset Harte and Gillen 2015, using Algorithm 2 of the optimisation-based method. 48

3.14 The loss plot for projecting a single video frame from the TCD-TIMIT dataset into the latent space of the StyleGAN generator, which was pre-trained on talking-head video frames from the same dataset, using the optimisation-based method described in Algorithm 2. 49

3.15 Visualisation of the video projection results using Algorithm 2 into the latent space of the StyleGAN generator pre-trained on video frames of talking heads from the TCD-TIMIT dataset. The visualisation is based on the first two principal components computed from the latent vectors obtained through the projection. 50

3.16 The pSp Encoder framework which we use to project video frames of talking heads. The *map2style* network is a fully convolutional network to gradually reduce the spatial size. Image is taken from Richardson et al. 2021 50

3.17 The results of projecting video frames from the TCD-TIMIT into the latent space of the StyleGAN model pre-trained on the FFHQ dataset using the pSp image encoder [Richardson et al. 2021]. 51

3.18 Visualisation of the video projection results using using the pSp image encoder [Richardson et al. 2021] into the latent space of the StyleGAN generator pre-trained on the FFHQ dataset. The visualisation is based on the first two principal components computed on the latent vectors obtained from the projection. 51

3.19 The framework for training an image encoder to project video frames into the latent space of the StyleGAN image generator. 52

3.20 The results of projecting video frames from the TCD-TIMIT dataset using an image encoder. The encoder is trained to map video frames of talking heads into the latent space of the StyleGAN image generator, which is pre-trained on the TCD-TIMIT dataset. 54

3.21 Visualisation of the video projection results using the image encoder into the latent space of the StyleGAN generator pre-trained on talking-head video frames. The visualisation is based on the first two principal components computed on the latent vectors obtained from the projection 54

3.22 Results of projecting video frames from unseen speakers into the latent space of a StyleGAN pre-trained on the TCD-TIMIT dataset during the training of the image encoder. The image encoder struggles to faithfully reconstruct the target video frames; however, it maintains consistent identity representation across the projections. 55

3.23 Overview of the unconditional video generation framework: Given an identity image I and a sequence of noise samples n_t, the aim is to generate a video of the given identity making typical facial expressions, such as in talking. 56

3.24 Video samples generated using the unconditional video generation model. The model employs an image encoder and a StyleGAN image generator pre-trained on video frames from talking-head datasets: (a) GRID and (b) TCD-TIMIT. We can observe that the model is capable of generating video frames that exhibit talking expressions. 60

3.25 Video samples were generated using the unconditional video generation model presented in Figure 3.23, with video discriminator loss 3.7. The model employs an image encoder and a StyleGAN image generator pre-trained on the FFHQ dataset of static head images. It is clear that the latent sequence generator struggles to produce convincing talking-like expressions, instead prioritising the enhancement of visual quality. 61

3.26 A modification of the framework presented in figure 3.23. The video discriminator takes as input the projected real video as well as the generated one. 62

3.27 Video samples generated using the unconditional video generation model trained with the image encoder E_I and the StyleGAN image generator G, pre-trained on the FFHQ dataset of static head images. The latent generator G_l and the video discriminator D_v are trained with the novel loss presented in Eq. 3.10 on video frames of talking heads. 63

3.28 Video samples were generated using the unconditional video generation model to demonstrate that the model learns to disentangle motion and appearance in the generated videos. The model is trained on video frames of talking heads with the image encoder and the StyleGAN image generator, which were pre-trained on images of static heads. 64

3.29 Video samples were generated using the unconditional video generation model with identity images taken from the wild. The model is capable of generating plausible videos featuring a variety of expressions. 65

3.30 The figure illustrates videos generated by our unconditional video generation framework in comparison with the MoCoGAN-HD model. It is evident from the figure that our framework can generate more diverse talking-like motions than the MoCoGAN-HD model, as highlighted in yellow. 67

4.1 Overview of the model: Given an identity image and a speech audio, the aim is to synthesise a video of the identity lip-synced with the audio. We first find the corresponding latent code of the identity image w_I using the image encoder E_I and then reduce its dimensionality using the latent encoder E_L. We then encode the audio using the audio encoder E_A. Next, we generate temporal noise n_t using the noise generator G_N. The decoder D then takes the reduced identity latent vector, encoded audio and the noise to predict a displacement to the identity latent vector in the latent space of a pre-trained image generator G. 72

4.2 Sampling strategy of audio segments and identity images in the first model. The identity images are randomly sampled from out of the audio sample window. The target frame is selected to be the fourth frame of the audio window. 76

4.3 Generated videos of our first model. We generated the videos using unseen speech audio from one of the training identities: (a) TCD-TIMIT and (b) GRID. For each sub-figure, the first row represents the ground truth frames of the input speech audio while the second row shows the generated frames 78

4.4 The outputs of our first model. We generated the videos using unseen audio inputs of unseen identities selected from two datasets: (a) TCD-TIMIT and (b) GRID. For each sub-figure, the first row represents the ground truth frames of the input speech audio. The other two rows show the generated frames. From this figure, our model synthesises videos well-synced with the input audio. It can be seen in the last column from each sub-figure that our model generates a non-open mouth when there is no spoken word 79

4.5 videos generated of unseen identities. We can see the first model fails to preserve the given identity. However, the generated videos have lip movements synced with the target video. 80

4.6 Overview of the model: Given an identity image and speech audio, the aim is to synthesise a video of the identity lip-synced with the audio. We first find the corresponding latent code of the identity image using the image encoder E_I. We then encode the audio using the audio encoder E_A. Next, we embed the identity latent code using the PCA basis V. The decoder D then takes both the embedded identity latent vector and encoded audio to predict a displacement to the identity latent vector in the latent space of a pre-trained image generator G. 82

4.7 This figure illustrates how the image encoder E_I can encode a video (top row) into the latent space of StyleGAN W^+ while preserving its mouth movements and other facial expressions. Bottom row shows the inverted video. 83

4.8 Sampling strategy of audio segments, identity images and the MFCC features in the second model. The identity images are randomly sampled from out of the audio sample window. The target frame is selected to be the middle frame of the audio window. 87

4.9 Generated videos of our second model. We generated the videos using the testing videos from two datasets: (a) GRID and (b) TCD-TIMIT. For each sub-figure, the first row represents the ground truth frames of the input speech audio while the second row shows the generated frames. From this figure, our second model synthesises videos well-synced with the input audio. 89

4.10 Samples generated using our approach. The top row shows frames from a source video providing the audio used to drive the generation. The middle row shows the corresponding generated frames where the generator G is tuned on a single frame. The bottom row shows generated frames where the generator G is tuned on a 5-second video clip. The figure demonstrates that tuning the generator with multiple frames leads to enhanced visual quality, as evident on the right side of the figure. These videos can be found at `https://mohammedalghamdi.github.io/talking-heads-acm-mm/` 90

4.11 The figure displays videos generated by the second model after Stage 1 and Stage 2 from two datasets: (a) TCD-TIMIT and (b) GRID. It is evident from the figure that the results of Stage 1 produce accurate lip movements; however, there are some distortions in the eyes and the skin colour, which are highlighted in red. In Stage 2, the results appear more realistic than in Stage 1, particularly in the appearance of the teeth, which are highlighted in yellow. 92

5.1 Qualitative comparisons: the figure shows video samples generated using our models in comparison with other state-of-the-art models on two benchmark datasets 100

5.2 Visual comparison of mouth-closure during bilabial events during the /p/ phoneme (highlighted in yellow). We can see that our models and the work of Vougioukas et al. produce a closed-mouth shape while the work of Prajwal et al. is out of sync. In addition, the work of Chen et al. and Zhou et al. fail to produce the required mouth shape. 101

5.3 Unseen speech audio. The figure shows video samples generated with speech audio from the wild. It can be seen that our second model generates accurate mouth movements in comparison with others (highlighted in yellow) 101

5.4 Unseen identity image. The figure shows video samples generated with an identity (top row) from the wild. It can be seen that our first model as expected did not preserve the identity image because the pre-trained StyleGAN image generator was trained on a few speakers 102
5.5 The interface of the user study shown to MTurk workers 103
5.6 User study results for evaluating our second model with two state-of-the-art models in terms of the audio-visual synchronisation (top) and the visual quality (bottom) 104

6.1 This figure shows samples of talking-head videos that were collected from the wild. These videos were aligned using the same method that was used to preprocess the FFHQ dataset for the StyleGAN model [Section 3.3.1.1]. 109
6.2 The figure demonstrates training the StyleGAN model on unaligned images results in unrealistic images. The upper sub-figure shows the real images of an unaligned training dataset used to train the StyleGAN model. The lower subfigure shows generated images from the trained StyleGAN. 110

A.1 The middle row shows video frames projected into the W^+ latent space with the use of Algorithm 1, while the bottom row displays the results of using Algorithm 2. The original video frames, shown in the first row, come from the GRID dataset. Visually, it can be seen that the quality of the video frames projected using Algorithm 2 is comparable to those produced by Algorithm 1. 113
A.2 The middle row shows video frames projected into the W^+ latent space with the use of Algorithm 1, while the bottom row displays the results of using Algorithm 2. The original video frames, shown in the first row, come from the TCD-TIMIT dataset. Visually, it can be seen that the quality of the video frames projected using Algorithm 2 is comparable to those produced by Algorithm 1. 113
A.3 Visualisation of the video projection results using Algorithm 1 and Algorithm 2, based on the first two principal components computed for each video. Algorithm 2 projects a video as a smooth trajectory within the latent space. The video sample is taken from the GRID dataset. 114

A.4 Visualisation of the video projection results using Algorithm 1 and Algorithm 2, based on the first two principal components computed for each video. Algorithm 2 projects a video as a smooth trajectory within the latent space. The video sample is taken from the TCD-TIMIT dataset. 114

List of Tables

2.1 Statistics of lab-based datasets and datasets from the wild. **Utter.**: number of utterances. 19

3.1 The table describes the image encoder architecture 53

3.2 The table describes the architecture of the video discriminator 57

4.1 The table describes the architecture of the audio encoder in the first model. The slope of Leaky ReLU is set to 0.02 in all layers. 73

4.2 The table describes the architecture of the latent encoder, noise generator, and decoder in the first model 73

4.3 The table describes the architecture of the audio encoder in the second model . . 84

4.4 The table describes the architecture of the latent decoder in the second model . . 85

5.1 Quantitative comparisons of our first and second models with four state-of-the-art methods on two datasets (TCD-TIMIT and GRID). **First** and <u>second</u> best scores for each measure per dataset. ↑ indicates a higher score is better while ↓ indicates a lower score is better. 97

5.2 Comparisons between our second model and others in terms of the number of parameters, inference time, and output size. 98

List of abbreviations

AV-CHMM	Audio-Visual Coupled Hidden Markov Model
CHMM	Coupled Hidden Markov Model
CNN	Convolutional Neural Network
EM	Expectation Maximization
FID	Frechet Inception Distance
GAN	Generative Adversarial Network
HMM	Hidden Markov Model
HMMI	Hidden Markov Model Inversion
LMD	Landmarks Distance Measure
LSTM	Long Short Term Memory
MFCC	Mel-frequency Cepstral Coefficient
MLP	Multi-layer Perceptron
MMM	Multidimensional Morphable Model
MSE	Mean Squared Error
PCA	Principal Component Analysis
PSNR	Peak Signal-to-Noise Ratio
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SSIM	Structural Similarity Index Measure
SVM	Support Vector Machine
VAE	Variational Autoencoder

List of symbols

a	Audio segment
D	Decoder network
d	Latent displacement
E	Encoder network
E_A	Audio encoder
E_I	Image encoder
E_L	Latent encoder
G	Image generator network
G_{noise}	Noise generator
G_l	Latent generator
I	Identity image of a speaker
M	Two-layer linear network
ML	Mouth Landmark points
n	Noise vector
p_{data}	Probability distribution of a training data
p_{model}	Probability distribution of a model
p_z	Prior probability
P	The number of ML points
v	Space formed by PCA components
W	Intermediate latent space
W^+	Extended intermediate latent space
x	Video frame
X	Video
z	Random vector sampled from a normal distribution

$\bar{w}$	Generated latent code
$\bar{x}$	Generated video frame
$\bar{X}$	Generated video
$\hat{x}$	Reconstructed/inverted video frame
μ	Mean of univariate variable
$\boldsymbol{\mu}$	Mean of multivariate variable
σ	Variance of univariate variable
Σ	Covariance matrix of multivariate variable
ϕ	Pre-trained neural network e.g. VGG [Simonyan and Zisserman 2015]

Chapter 1

Introduction

1.1 Introduction

The task of video generation has been an interest of the computer vision research community for the last three decades. The goal is to build a model that learns from existing videos and generates a novel video that closely resembles the training videos. Since videos are temporal data consisting of diverse motions, video generative models have to implicitly learn the underlying dynamics that drive these motions, thus imposing additional challenges, unlike the image generation task. These models could be driven by other signals such as text, audio and video. The choice of driving signal affects the complexity of the model. Synthesising videos of a talking head driven by a video is less difficult than by speech audio. For example, cross-identity facial re-enactment where the task is to transfer facial expressions, lip movements, pose and gaze from a driving video to an image identity of a target speaker. These visual cues provided by the driving video can aid in understanding and synthesizing coherent output video. Some video generative models are driven by more than one signal. For example, one can drive lip movements of an identity image using speech audio, while pose and other facial expressions are driven by a video. In this thesis, we focus on the task of video generation of talking heads driven by speech audio.

Synthesizing talking head videos using speech audio involves generating a realistic video of a speaker lip-synced with the given speech. The video generative model takes as input one or a few identity images of the target speaker and speech audio. Figure 1.1 defines the task of synthesising videos of a talking head from speech audio. The model analyses the speech audio

Figure 1.1: The task of audio-driven talking-head video generation. The generative model takes an identity image and speech audio as inputs and generates a video of the identity, lip-synced with the given speech audio.

to determine the precise mouth movements while the identity image is used to determine the visual appearance of the talking head.

Synthesizing talking head videos is a challenging task due to several factors.

Realism. A primary challenge of synthesising talking-head videos from speech audio is to generate a video that looks photo-realistic and indistinguishable from real ones. Generating talking-head videos with accurate mouth movements that match the speech audio requires a model to understand the relationship between the audio and facial expressions. A talking head video is made up of various factors: the unique traits of the person, the movements of the head, the angle of the camera and the facial expressions. These factors define the deformation of facial features depicting a talking head [Chen et al. 2020]. A good video generative model needs to generate temporally coherent video with high visual quality. Human eyes are sensitive to temporal discontinuities in videos, just as they are sensitive to spatial discontinuities in images.

Several works [Garrido et al. 2015; Suwajanakorn et al. 2017b; Thies et al. 2020a] solve this challenge by editing the mouth region of an existing target video in accordance with the given speech audio. These approaches are graphic-based approaches which require access to extensive footage of the target speaker. Although they generate highly photo-realistic talking-head videos, they are subject-dependent and cannot generalise to a speaker from the wild. Suwajanakorn et al. 2017b use a recurrent neural network (RNN) to generate mouth landmarks from speech audio. Using the mouth landmarks, they generate the mouth texture and compose it into the mouth region of a target video. The model is trained with 17 hours of footage of a single subject and produces realistic talking-head videos from speech audio.

Complexity. Another primary challenge is the learning complexity of the generative model. The task of synthesising talking-head video from speech audio is a multimodel problem. The learning process involves learning the mapping between the speech audio and the mouth movements to generate accurate motion and dynamics while also learning to generate the accurate visual structure of the head. A common approach to mitigate this task is to split the learning process into two steps: one to learn the audio alignment to facial characteristics e.g. facial landmarks and the other to learn to synthesise the video from the facial characteristics. In the era before deep learning, most of the research works [Yamamoto et al. 1998; Brand 1999; Choi et al. 2001a; Xie and Z.-Q. Liu 2007b] use machine learning algorithms such as Hidden Markov Models (HMMs) to learn the underlying dynamics of the audio alignment and then use standard computer graphics algorithms to synthesise the videos. For the last decade, the research community has used RNNs to learn audio alignment while using convolutional neural networks (CNNs) to generate videos.

The works of [Chen et al. 2019a; Y. Zhou et al. 2020b; Das et al. 2020] learn to map speech audio to 2d intermediate features such as facial landmarks to generate a sequence of facial landmark vectors that drives the lip movements of a talking head. Other works [Karras et al. 2017a; Yi et al. 2020; Ji et al. 2021] use speech audio to drive the 3d vertex positions of a face model. Then, a deep learning rendering network is used to generate the talking-head video based on the updated 3d vertex positions.

Generalisation. Producing a talking-head generative model that works on unseen identity is challenging. A generic model requires a large dataset of talking-head videos to be trained on. Such a dataset was difficult to acquire until the internet provided an abundant source of videos. Chung et al. 2018 collected a large talking-head dataset by downloading YouTube videos of famous people. Simply training a model on large datasets does not in itself lead to the ability to generalise. However, one needs to create a suitable model to utilise large datasets. Chung et al. 2017a introduce a generic model that encodes an identity image and audio speech using two separate encoder networks and then a decoder network generates a single frame video from a concatenation of the image and audio embeddings. Although the model successfully generalises to unseen subjects, the resulting videos are typically limited in terms of overall realism and resolution. Many previous generic models generate low-resolution video [Vougioukas et al. 2018a; Vougioukas et al. 2019] or cropped faces [Chung et al. 2017a; Chen et al. 2019a]. Low-resolution video is generally not suitable for deployment in many real-world applications, such as virtual assistants.

1.2 Applications

The ability to synthesise talking-head videos has many useful applications. One application of talking-head videos is in the entertainment industry. AI-generated video technology will create an engaging and immersive experience in games. For example, the technology allows players to interact in real-time with characters instead of playing pre-rendered videos of the characters, which are less engaging [Parke and Waters 2008]. Another example is in film production, where one can use AI-generated video dubbing in another language. Additionally, the technology can be used to recreate realistic scenes of deceased actors or scenes featuring new characters.

Another application is in education and training. The technology can be used to create personalised educational videos or tutorials interactively in a more engaging way. This can benefit in improving the skill sets of individual students. It can also be used to generate training videos for employees in different languages in a multinational company. Additionally, there are other applications such as virtual assistants, advertising videos and video teleconferencing.

1.3 Research motivation

Recent advances in image synthesis have been successful in generating high-resolution images from a random vector [Karras et al. 2017b; Brock et al. 2019; Karras et al. 2019]. Karras et al. 2019 propose a style-based generator, StyleGAN, that synthesises high-quality images that are largely indistinguishable from real ones. The StyleGAN model consists of two components: a mapping network that maps a random vector sampled from a normal distribution to an intermediate latent space and a synthesis network that uses the "styles" output of the mapping network as input after each of its convolutional layers to generate new images. Karras et al. 2019 show that the styles in the intermediate space are not correlated, meaning that each subset of styles controls meaningful high-level attributes such as the face shape of generated images. Some works have further studied the intermediate latent space of a pre-trained StyleGAN image generator [Roich et al. 2021; Abdal et al. 2019a; Y. Shen et al. 2020; Härkönen et al. 2020] and discovered meaningful semantics for manipulating images, such as changing the gender of a

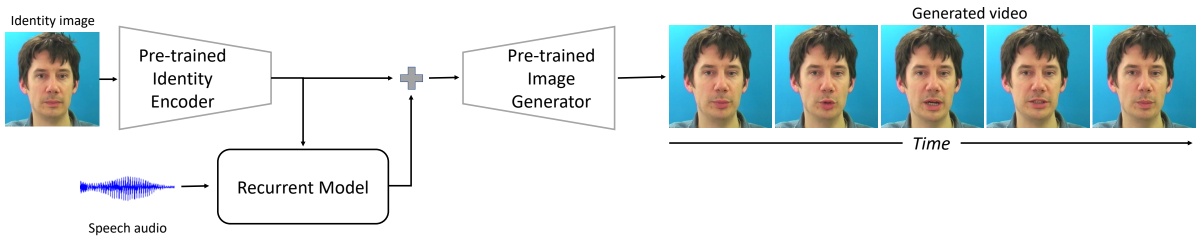


Figure 1.2: An overview of our novel approach to synthesise talking-head videos from an identity image and speech audio.

generated face image or making a face smile.

Inspired by these works, we propose a novel approach for generating high-resolution videos conditioned on speech audio by constructing trajectories in the latent space of a pre-trained StyleGAN image generator [Karras et al. 2019]. The approach is to find the latent code of a given identity image in the latent space and a generative model displaces the latent code recurrently conditioned on speech audio. Our approach uses a pre-trained image encoder to find the latent code of a given identity image in the latent space of the generator. We then train a recurrent audio encoder along with a latent decoder to predict a sequence of latent displacements to the encoded identity image. The approach reduces the complexity of generating talking-head videos by utilising a pre-trained StyleGAN image generator while generating photo-realistic high-resolution videos from any given identity image using novel speech audio. Figure 1.2 gives an overview of our approach.

We start by studying two GAN-inversion methods to project videos into the latent space of a pre-trained StyleGAN image generator. We then propose a video generative model to synthesise talking-head videos from random samples to demonstrate that the encoder-based method for GAN inversion is suitable for video generation. We then implement our audio-driven talking-head generation approach with two models: The first is using the latent space of a StyleGAN image generator pre-trained on video frames of a talking head and the second is using a StyleGAN image generator pre-trained on faces with static expressions. We also show that the second model can generate high-resolution videos of any identity from the wild. We compare our two models with other state-of-the-art approaches qualitatively and quantitatively using benchmark measures: LMD, SSIM, PSNR and FID. Our results show that our approach outperforms the state of the art models on two commonly used datasets; TCD-TIMIT [Harte and Gillen 2015] and GRID [Cooke et al. 2006].

1.4 Contributions

Our contributions are as follows:

- In Chapter 3, we develop an encoder-based projection approach to project videos into the latent space of the StyleGAN image generator. This approach guarantees smoothness in the latent space and faithfully reconstructs the input videos.
- In Chapter 3, we propose an unconditional video generation approach that utilises the intermediate latent space of the StyleGAN image generator. Unlike previous works, our approach employs an image encoder to ensure the preservation of the given identity and to facilitate the training of the model.
- In Chapter 4, we propose a generative approach that generates talking-head videos driven by speech audio. This approach is built on top of a pre-trained StyleGAN image generator by using an encoder-based projection method to find the latent code of an identity image. We then use speech audio to drive this code in the latent space. We experiment with two models: one is with a StyleGAN image generator pre-trained on a dataset of human faces with no expressions and the other is with a StyleGAN image generator pre-trained on video frames of the talking-head dataset. The latter allows our approach to generate high-resolution videos and generalise to any identity from the wild.
- In Chapter 5, we evaluate the performance of our audio-driven talking-head models both quantitatively and qualitatively, comparing them to other state-of-the-art models using benchmark measures such as LMD, SSIM, PSNR, and FID. Additionally, this includes assessing their performance in various scenarios, including bilabial events and unseen identities, which are not commonly addressed in recent research papers. We also conducte a human user study to gather subjective feedback on the performance of the models, which is essential.

1.5 Publications

The research conducted for this thesis has resulted in a peer-reviewed publication listed below:

- Mohammed M. Alghamdi, He Wang, Andrew J. Bulpitt, and David C. Hogg. **"Talking Head from Speech Audio using a Pre-trained Image Generator"** In Proceedings of the 30th ACM International Conference on Multimedia, 2022. Alghamdi et al. 2022

Parts of the thesis have been presented in the following workshop:

- Mohammed M. Alghamdi, He Wang, Andrew J. Bulpitt, and David C. Hogg. **”Video Synthesis of a Talking Head”** The British Machine Vision Association Symposium, 2022.

1.6 Thesis outline

This thesis is organised into six chapters. In Chapter 2, we review the state-of-the-art models in the literature of audio-driven talking-head video generation. We also review relevant literature in image synthesis using GANs, GAN-inversion and video generation.

In Chapter 3, we explore two GAN-inversion methods for projecting a video in the latent space of a pre-trained StyleGAN image generator. We also use the learned projection to build a GAN-based generative model to generate videos from noise samples using a pre-trained StyleGAN image generator.

In chapter 4, we propose a novel approach that synthesises a talking-head video from speech audio using a pre-trained StyleGAN image generator. We experiment with two models: the first uses a StyleGAN image generator pre-trained on video frames of a talking head and the second uses a StyleGAN image generator pre-trained on faces with static expressions (i.e., not talking). For both models, we demonstrate the training strategy and show the video samples generated using the approach.

In Chapter 5, we run a thorough evaluation comparing our two models, presented in Chapter 4, with the state-of-the-art models. We introduce quantitative measures for evaluating talking-head videos and discussing the results. Additionally, we also show the visual results of our models with other models in different scenarios.

Chapter 6 concludes the thesis with a discussion of the limitations of the proposed approach and suggestions for future work.

Chapter 2

Background

This chapter provides an overview of the development of synthesising talking-head videos and other related methods. Specifically, we review existing methods in synthesising talking-head videos from speech audio and explore various available talking-head datasets. We also review recent advancements in image synthesis using Generative Adversarial Networks (GANs), with a focus on the StyleGAN model [Karras et al. 2019]. Furthermore, We discuss GAN inversion methods for projecting images in the latent space of a pre-trained StyleGAN model. Additionally, we discuss methods for understanding the latent space and show their results. Finally, we explore current methods for unconditional video generation, particularly those based on the StyleGAN model.

2.1 Audio-driven talking-head videos

Synthesising talking-head videos has long been a research challenge in the computer vision community. Prior to deep learning, most of the approaches can be categorised into two approaches: learning-based methods or sample-based methods [Z. Deng and Noh 2008]. The learning-based methods learn a statistical model over the training samples. On the contrary, sample-based methods use a sequence of face images to reconstruct the talking-head model. In the following, we review both methods as well as the recent deep learning methods.

2.1.1 Learning-based methods

Prior to deep learning, most of the learning-based methods use the hidden Markov model (HMM) to generate talking-head videos from audio. In the 1990s, the HMM was widely used in speech recognition applications [Rabiner 1989]. The HMM is a probabilistic approach where the system being modelled is considered to be a Markov process with "hidden" states that cannot be observed from data.

Yamamoto et al. 1998 was the first to apply HMM to audio-visual mapping in visual speech synthesis tasks. The approach is to learn the HMM from audio data and use the Viterbi algorithm [Forney 1973] to align each video sequence with the audio sequence. Brand 1999 proposes Voice Puppetry, an alternative HMM-based approach to solve the task. Brand assumes that there is mutual information between vocal and facial gestures. First, the model uses the Baum-Welch algorithm to estimate facial HMM parameter values from the training video. In this training stage, the model learns a mapping from each hidden state to facial parameter space and a state probabilities matrix for each video frame. Second, the model reuses the learned facial HMM to construct a vocal HMM resulting in a dual mapping into acoustic feature space. Given new audio, the model uses the Viterbi algorithm on the vocal HMM to compute the most probable sequence of facial states. One major weakness of these approaches is the use of Viterbi which is not robust to noise in the audio and may represent only a small fraction of the total probability mass [Fu et al. 2005]. Choi et al. 2001b propose an HMM inversion (HMMI) to avoid the use of the Viterbi algorithm in synthesis by instead using the expectation maximization (EM) algorithm. Xie and Z.-Q. Liu 2007a adopt a similar approach to explicitly model the characteristic of bimodal speech (e.g. asynchrony and synchrony) using a coupled HMM (CHMM). The training of audio-visual CHMM (AV-CHMM) is done offline. Given a novel audio and trained AV-CHMM, the method uses an EM algorithm to compute the optimal visual parameters. This is in contrast to Brand's approach [Brand 1999] where the Viterbi algorithm is used.

2.1.2 Sample-based methods

Bregler et al. 1997 propose a method which uses existing footage of a speaker to synthesise a new video given speech audio. In the analysis stage, the method automatically creates a video database by labelling the audio track of the training talking-head videos into phonemes.

The phonemes are then used to segment the training videos into short video clips, each having three phonemes or a triphone. These triphone videos, accompanied by the fiduciary points of the mouth and jaw, along with the phoneme labels are stored within the video database. In the synthesis stage, given new audio, background video and a video database, the method segments the audio and searches for appropriate triphone video sequences in the database. The triphone videos are then blended into the background video. The use of triphone sequences is to handle coarticulation effects which therefore requires a large number of training videos to build a database with all possible triphone sequences. This requirement is needed to generate photo-realistic videos on a given novel audio. Ezzat et al. 2002 propose another approach which overcomes the requirement of large training samples. The approach is based on a multidimensional morphable model (MMM) to generate new lip sequences given limited training samples. It models mouth images into a multidimensional space where its axes are the mouth's appearance and shape. Given new audio, the approach synthesises a smooth trajectory in the MMM space.

2.1.3 Deep learning methods

Although an HMM is efficient in learning sequential audio-visual data, it has some limitations, such as modelling long dependencies and handling high-dimensional data. In this section, we review deep learning methods for synthesising talking-head videos, which are known to handle high-dimensional data and model long dependencies. B. Fan et al. 2015 use recurrent neural networks (RNNs) to predict visual features of the lower face from contextual phoneme labels of a given audio. The predicted visual features are used to reconstruct the texture of the lower face which is then blended to a background face. The RNN-based model outperforms the HMM-based models in generating high-quality talking-head videos. Charles et al. 2016 adopts a similar approach by using an LSTM-based network to map phonetic labels to visemes which represents the visual mouth motion. To ensure smooth visual transitions, the Principal Component Analysis (PCA) coefficients of the last frame are enforced to match the ones of the next frame using the Viterbi algorithm [Forney 1973]. Similarly, Suwajanakorn et al. 2017b predict the landmark points of the mouth sequentially using an RNN network driven by the Mel-frequency cepstral coefficients (MFCC) of audio segments. The output mouth landmarks are then used to synthesise the mouth texture, followed by a post-processing step of video retiming and jaw smoothing before compositing it into a background video of the target speaker.

The aforementioned approaches work only on a single subject and require a large amount of footage of the target speaker. In addition, they only synthesise the mouth region and blend it back into a target video.

With the existence of large taking head datasets [Afouras et al. 2018; Nagrani et al. 2020; Harte and Gillen 2015; Cooke et al. 2006], numerous methods [Chung et al. 2017a; K R Prajwal et al. 2020a; Vougioukas et al. 2018a; Chen et al. 2018; Y. Song et al. 2019; H. Zhou et al. 2019] have been proposed to build a generic talking-head model. Chung et al. 2017b propose an approach that generates videos using multiple images of a target face and speech audio. The model consists of an audio encoder and identity encoder that learn a joint embedding of the face and audio, respectively and an image decoder that generates a frame that best represents the given audio for the target face. The model is trained in an end-to-end training strategy with L_1 loss which is known to produce blurry images; therefore, a pre-trained generic deblurring network was used to enhance the visual quality of generated videos. K R Prajwal et al. 2020a adopt a similar approach, except the identity encoder takes as input the target video (lower half masked) as well as multiple identity images. The approach uses a visual discriminator loss in addition to L_1 loss to improve the visual quality. To enhance the quality of the lip-synchronisation, the approach uses a pre-trained lip-sync discriminator which has been trained on a very large lip reading dataset [Afouras et al. 2018]. Vougioukas et al. 2018a expand the approach by training a recurrent-based decoder with a noise generator to model spontaneous face expressions, e.g. blinks. The model is trained with a sequence discriminator and an audio-visual discriminator as adversarial losses. The previous models are trained on large multi-speaker talking-head datasets to be able to generate a talking-head video of an arbitrary target speaker.

Other models rely on 2D intermediate features e.g. facial landmarks to learn the mapping between audio input and video output [Chen et al. 2019a; Y. Zhou et al. 2020b; Das et al. 2020]. Chen et al. 2019a propose a cascade approach that generates a talking face video given an image and audio. The method first transfers the given audio signal to facial landmarks and then generates video frames conditioned on the landmarks. Y. Zhou et al. 2020b adopt a similar approach that first disentangles the content and speaker information from the input audio. Then, these two components are mapped to content and speaker facial landmark spaces using a recurrent model on each. Figure 2.1 shows videos generated using some of the aforementioned

Figure 2.1: Figure shows generated videos using five different methods [Chung et al. 2017a; Vougioukas et al. 2018b; Chen et al. 2019b; Y. Zhou et al. 2020a; K R Prajwal et al. 2020a]

models.

3d facial animation. These methods [Taylor et al. 2017; Karras et al. 2017a; Cudeiro et al. 2019; Thies et al. 2020a; Richard et al. 2021a; Y. Fan et al. 2022] build deep learning models that learn to drive a 3D face model using speech audio. Karras et al. 2017a build a convolutional model that maps audio waveforms to per-vertex displacements from the neutral pose of the face mesh. Their approach is subject-dependent and can learn different speaking styles and facial expressions of speakers. Cudeiro et al. 2019 extend the approach to generalise to new speakers by factoring identity from facial motion and representing audio using low-dimensional features extracted using the DeepSpeech model [Hannun et al. 2014]. Although these models learn to successfully animate a 3D face model from audio, the facial motions are only present in the mouth region. Richard et al. 2021b learn to disentangle audio-correlated and audio-uncorrelated information by using a categorical latent space. For example, eye blinks should not be correlated to specific lip movements. The goal is to generate plausible upper-face motions independent of the input audio using an autoregressive model that samples motions from the learned categorical space.

Transformer-based talking-head Several works [Y. Fan et al. 2022; Zhong et al. 2023; S. Wang et al. 2021; J. Wang et al. 2023] have used transformers [Vaswani et al. 2017] instead of RNN units to learn the temporal dynamics of talking-head videos. Zhong et al. 2023 utilizes a transformer network to generate the landmarks of lips and jaw from speech audio. In this context, the transformer is used to learn the temporal relationship between the speech audio and the landmarks, similar to the approach of Chen et al. 2019a, which employs LSTM units to predict landmarks from speech audio. Zhong et al. 2023 found that transformer-based networks outperform their LSTM-based ones. Similarly, Y. Fan et al. 2022 employs transformers to generate a sequence of animated 3D face meshes. One advantage of using transformers over LSTMs is that training transformer-based models is faster due to their ability to be executed in parallel, although they have the same inference time as they run autoregressively.

Diffusion-based talking-head Yu et al. 2022 were the first to introduce the use of diffusion models for the task of audio-driven talking-head generation. They trained a diffusion model that takes as input a sequence of audio representations and a sequence of noise vectors to generate non-lip facial motions such as pose, expression, blink, and gaze in the form of feature vectors. The method then utilizes a GAN generator to synthesise the talking-head video frames. This approach replaces the traditional autoregressive method for generating facial motions with a probabilistic diffusion model capable of producing diverse motions given the same audio input. S. Shen et al. 2023 extended this approach by conditioning the diffusion model on audio features, facial landmark features, and reference image features to generate a sequence of features. These features are then used as input to a pre-trained image generator to synthesise the talking-head video frames. Stypułkowski et al. 2023 introduced a method for directly generating video frames using a frame-based diffusion model conditioned on audio features, motion frames, and noise vectors. While these models successfully generate high-resolution talking-head videos, the inference time remains significant due to the denoising process that is inherent in diffusion models.

2.2 Talking-head datasets

There have been numerous datasets of talking heads over the past two decades. Mainly, they are lab-based datasets where actors are recruited to be recorded uttering sentences [Cooke et al. 2006; Harte and Gillen 2015; Cao et al. 2014]. These datasets are recorded under controlled conditions and require expensive resources. Recently, datasets have been collected from the wild, mostly from YouTube websites [Suwajanakorn et al. 2017b; Nagrani et al. 2020; Z. Zhang et al. 2021]. These datasets are cost-effective and easily obtainable from video sources provided by the internet e.g. YouTube. In the following two subsections, we explain both categories in detail.

2.2.1 Lab-based datasets

Lab-based talking-head datasets are recorded with a green screen (Chroma Keying) and under controlled light sources. Cooke et al. 2006 built GRID, a dataset of 34 speakers, each uttering 1000 sentences of 6 words from a selected vocabulary. These videos were recorded with a frontal view. Harte and Gillen 2015 recorded the TCD-TIMIT dataset with 59 speakers, each uttering 100 sentences. The sentences were selected from the TIMIT corpus [Garofolo 1993]. The videos of GRID and TCD-TIMIT datasets are recorded with neutral expressions and no emotions. Cao et al. 2014 built CREMA-D an emotional dataset where actors utter sentences with six different emotions e.g. anger, surprise, etc. K. Wang et al. 2020 propose MEAD a similar emotional talking-head dataset but with higher resolution and seven different views. The sentences were based on the TIMIT corpus. Figure 2.2 shows video samples selected from the lab-based datasets.

2.2.2 Datasets from the wild

Datasets collected from the wild have more variation than lab-based ones. In talking-head videos, for example, they have different poses and backgrounds, and their associated speech audios are often corrupted with noise e.g. music. Therefore, these types of datasets are very challenging to model. Suwajanakorn et al. 2017b collected a dataset of a single speaker, President Barack Obama, from his weekly presidential addresses videos. Nagrani et al. 2020 propose the VoxCeleb dataset which contains over 1 million video clips extracted from YouTube videos. The dataset has two versions: VoxCeleb1 and VoxCeleb2. Both combined have around 7000

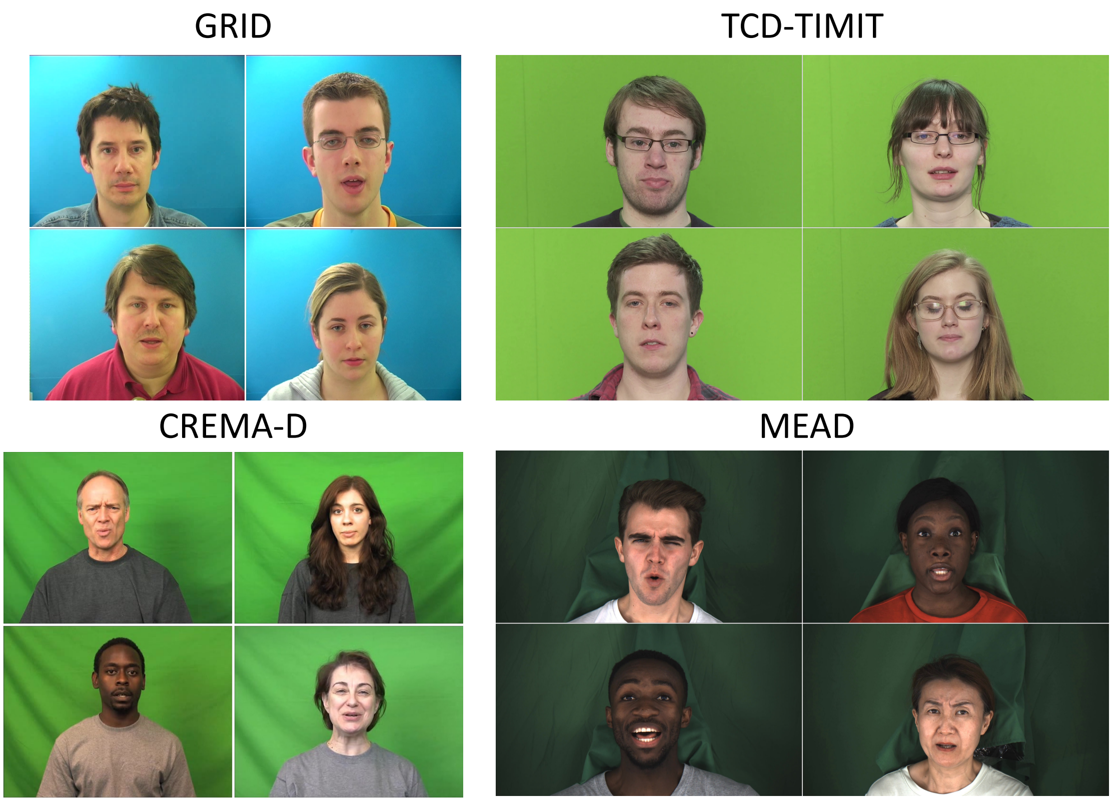


Figure 2.2: Samples selected from lab-based datasets

speakers and the spatial dimension of each video is 224 x 224. Z. Zhang et al. 2021 propose HDTF a smaller video dataset but with higher resolution. The dataset consists of 370 different videos of politicians from the United States. Figure 2.3 shows sample frames from the above datasets. The statistics of datasets described in both categories are presented in Table 2.1.

2.3 Generative Adversarial Network (GAN)

Although the basic idea of GAN was only introduced a few years ago, it has been the most popular approach to generative models. A GAN works without explicit density estimation and its idea is based on game theory where there are two players, a **generator** G and a **discriminator** D. The generator learns to generate samples similar to the training samples and the discriminator decides whether the samples are real or fake. Real samples come from the training data and fake samples are generated by the generator. The goal of the generator is to fool the discriminator by creating samples which are intended to be drawn from the underlying distribution of the training samples.

The generator is a differentiable function $G(z; \theta_g)$ that learns the mapping between an input

Figure 2.3: Samples selected from datasets from the wild

Dataset	# of speakers	# of utter.	# of emotions	Resolution
Lab-based datasets				
GRID [Cooke et al. 2006]	34	34,000	-	720x576
TCD-TIMIT [Harte and Gillen 2015]	59	5,782	-	1920x1080
CREMA-D [Cao et al. 2014]	91	7,442	6	1280x720
MEAD [K. Wang et al. 2020]	60	281,400	8	1920x1080
Datasets from the wild				
Voxceleb [Nagrani et al. 2020]	7,363	1,281,762	-	224x224
Obama [Suwajanakorn et al. 2017a]	1	304	-	1920x740
HDTF [Z. Zhang et al. 2021]	276	370	-	1280x720

Table 2.1: Statistics of lab-based datasets and datasets from the wild. **Utter.**: number of utterances.

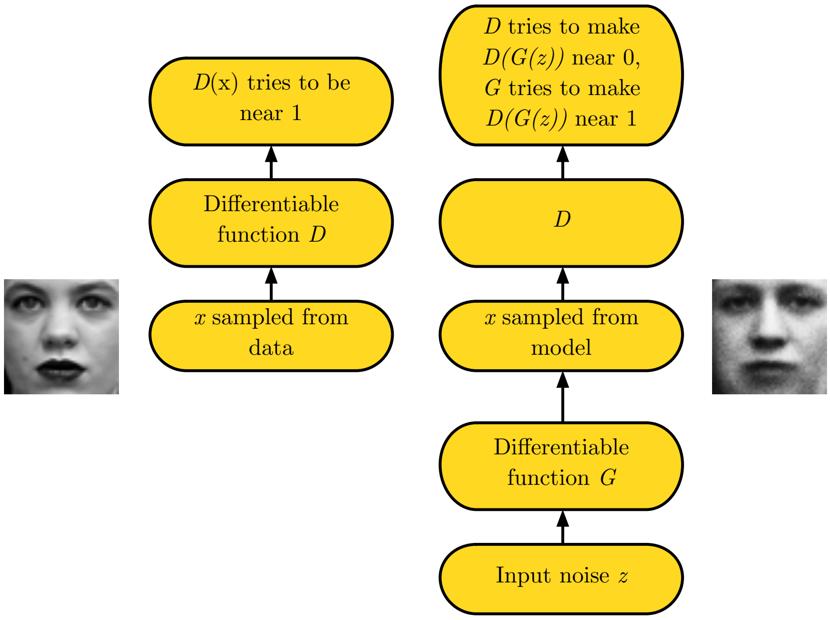


Figure 2.4: The GAN framework, figure taken from Goodfellow 2016

noise variable z, sampled from a prior distribution $p_z(z)$, and the data space. When z is sampled from, for example, a Gaussian distribution, the function G yields a sample of x drawn from p_{model} and is represented by a deep neural network model with parameters θ_g. The discriminator is also a differentiable function $D(x; \theta_d)$ that takes an input x and outputs the probability that x came from the data distribution p_{data} rather than the generator's distribution p_{model}. $D(x; \theta_d)$ is represented by a deep neural network model with θ_d [Goodfellow et al. 2014]. The GAN framework is illustrated in Figure 2.4.

The discriminator takes samples drawn from p_{data} and p_{model}, and is trained to maximize the probability of classifying the samples correctly. The generator is trained simultaneously to minimize the log probability of the discriminator being correct. Therefore, the two models play a minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{2.1}$$

The procedure of training GAN consists of simultaneous Stochastic Gradient Descent (SGD) application on both models and uses an iterative approach. On each step, we sample a minibatch of x from the training data and a minibatch of x drawn from p_{model}. We update the discriminator's parameters θ_d and the generator's parameters θ_g by their respective stochastic gradients. Figure 2.5a shows images generated using the GAN model [Goodfellow et al. 2014].

2.4 Image synthesis using GANs

Image synthesis has seen rapid improvement recently. The most prominent approaches are Generative Adversarial Networks (GANs) [Goodfellow et al. 2014], Variational Autoencoders (VAEs) [Kingma and Welling 2013], and diffusion models [Sohl-Dickstein et al. 2015]. VAEs approximate the distribution of the training images by encoding a high-dimensional input image into a lower-dimensional latent representation and then decoding it back. This process minimizes the Kullback-Leibler divergence between the approximate distribution of the VAE and the true data distribution of the training images. On the other hand, diffusion models define a Markov chain of diffusion steps that gradually add noise to the training images and then learn to reverse this process. Unlike VAEs, diffusion models follow a predetermined procedure with known noise levels at each step, and the latent variable of diffusion models shares the same high dimensionality as the training image.

GANs and diffusion models are known for producing higher-quality images than VAEs, while diffusion models generate more diverse images than GANs. However, diffusion models are slower during inference due to the iterative noise removal process, posing challenges for their integration as the image generator in tasks such as talking-head video generation. Here, we will focus on image synthesis using only GANs.

Since the original GAN paper was published, there have been several different approaches proposed to improve the quality of the generated samples. Radford et al. 2016 propose Deep Convolutional Generative Adversarial Networks (DCGAN) to syntheise images from a noise sample. The model relies on only a fully-convolutional network to synthesise images with a resolution of 64x64. The use of convolutional layers in in the generator and discriminator networks allows the model to capture hierarchical patterns in images. In addition, they introduce the use of batch normalization in both networks and use the LeakyReLU activation in the discriminator's layers. Figure 2.5b shows images generated using the DCGAN model.

Karras et al. 2018 extends the approach to synthesising high-resolution images by introducing ProgressiveGAN, a new training methodology. The main idea is to grow the size of the generator and discriminator as training progresses, adding layers to increase the resolution of the generated images. Instead of starting with a large network, the approach begins with a smaller network that generates low-resolution images (e.g., 4x4). As training progresses, they approach keeps

(a) Images generated using the original GAN model, adapted from [Goodfellow et al. 2014]

(b) Images generated using the DCGAN GAN model, adapted from [Radford et al. 2016]

Figure 2.5: The figure shows images generated using the original GAN model and the DCGAN model

adding layers to both networks, thereby increasing the resolution of the generated images. This allows the model to effectively understand low-level details from early training stage and then gradually learn high-level details as the model keeps growing. Figure 2.7a shows images generated using the ProgressiveGAN model.

2.4.1 Style-based Generator: StyleGAN

Karras et al. 2019 propose StyleGAN, a style-based generator motivated by the style transfer literature. In the new design, the latent code (noise variable) is provided to the generator as styles compared to the traditional generator which feeds the latent code through the first layer. Given a latent code z from Z space, a mapping network is learned to map from Z to W space. For simplicity, the dimensionality of both spaces is set to 512 and the mapping network is 8 fully-connected layers with nonlinear activation functions. The intermediate latent space W controls the generator using adaptive instance normalization (AdaIN) operation at each convolution layer with a learned affine transformation. This localises the affect of each style in the synthesis network, thus enabling the network to implicitly learn a separation of high-level attributes, such as pose and identity in an unsupervised manner. As a result, the intermediate latent space of the network becomes disentangled, which contrasts with the previous traditional approach where the noise variable was passed through a single layer, leading to an entangled latent space. Figure 2.6 shows an overview of the StyleGAN generator. The model can generate high-resolution images (Figure 2.7b).

2.5 GAN inversion

Since GAN models have been successful at synthesising photo-realistic images, the latent space of these models encodes rich and semantic information. The goal of GAN inversion is to find a latent code of a given image in the latent space of a pre-trained GAN generator to fully reconstruct the image. This capability can extend the use of these GAN models to various applications, including real image editing and manipulation. It also provides insight into how these GAN models work [Xia et al. 2021].

To design a GAN inversion method, one has to set the desired properties of such a method. There are two properties of GAN inversion, (1) the ability to faithfully reconstruct the given image and (2) the ability to facilitate downstream tasks [Xia et al. 2022]. Suppose we have a

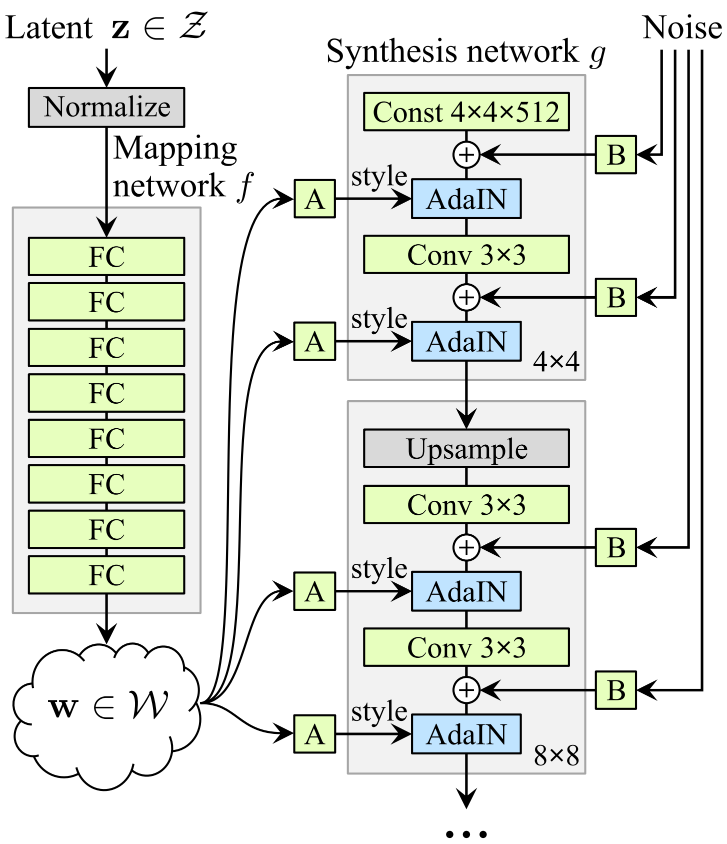


Figure 2.6: The StyleGAN generator architecture, figure adapted from Karras et al. 2019. Here "A" stands for a learned affine transformation.

(a) Images generated using the ProgressiveGAN model, adapted from [Karras et al. 2018]

(b) Images generated using the StyleGAN model, adapted from [Karras et al. 2019]

Figure 2.7: The figure shows images generated using the ProgressiveGAN model and the StyleGAN model

generator G that takes a latent vector z sampled from a normal distribution Z and generates an image x in the image space, the goal of GAN inversion is to find $\bar{z}$ that produces an image $G(\bar{z})$ similar to x. The problem of GAN inversion is defined as follows:

$$\bar{z} = \arg\min_{z} \ell(G(z), x), \tag{2.2}$$

where ℓ is a distance function that measures how close the original image x to the reconstructed image $G(\bar{z})$ from the back-projection of x into the latent space.

In the literature on GAN inversion, most of the methods aim to invert the StyleGAN network due to its prevalence in the image synthesis domain. These methods can be categorised into three categories: encoder-based method, optimization-based method and hybrid-based method. The encoder-based method defines an encoder network that maps an image from the image space to the latent space of a pre-trained GAN generator. The optimisation-based method solves the problem by optimizing the latent code directly via back-propagation. The hybrid-based method combines both previous methods to solve the problem. In the following, we review these three methods.

2.5.1 Encoder-based GAN inversion

This method solves Eqn. 2.2 by training an encoder E to encode a given image X in the latent space Z of a pre-trained generator G. The training is done in an auto-encoder fashion where the aim is to reconstruct an image that is as good as the original image. However, the encoder E is only trained while the pre-trained generator G is kept fixed. This method solves the following objective function:

$$\bar{z} = \arg\min_{z} \ell(G(E(x; \theta^{*})), x), \tag{2.3}$$

There are several GAN inversion works [J.-Y. Zhu et al. 2018; Richardson et al. 2021; Alaluf et al. 2021; Wei et al. 2022] that use an encoder-based approach. J.-Y. Zhu et al. 2018 was an early work that proposed to invert GAN by training a feed-forward encoder to map an image to the latent space of the DCGAN network [Radford et al. 2016]. The objective was to minimise the feature loss between the input image and the reconstructed image. Richardson et al. 2021 propose a similar approach by designing an encoder that inverts a pre-trained StyleGAN network. The model called pSp, is trained to map real images to the extended latent space W^+. The pSp encoder is then utilised for image-to-image translation tasks. Alaluf et al. 2021 improves further the reconstruction performance by introducing an iterative inversion scheme, ReStyle. The scheme starts by iteratively encoding a concatenation of a given input image x along with its previous reconstructed image $\bar{x}_{t-1}$. The initial image $\bar{x}_0$ is internalised using the image of the average latent code. One disadvantage of this approach is slow compared to the pSp model.

2.5.2 Optimization-based GAN inversion

This method solves Eqn. 2.2 by optimising the latent vector z using gradient-based approach. The approach starts with an initialised z_0 and iteratively optimises it by computing a similarity loss between the input image and the reconstructed image. Several studies [Abdal et al. 2019b; Abdal et al. 2021; Xu et al. 2021; P. Zhu et al. 2021] have proposed optimisation-based methods for projecting a given image into the latent space of a pre-trained image generator. Abdal et al. 2019b study the method by inverting the pre-trained StyleGAN network on real images. They show that the method produces results that are superior to the encoder-based method. However, this method takes approximately 7 minutes to invert a single image whereas the encoder-based method takes a fraction of a second since it is a single feed-forward step. In addition, the optimisation-based method is sensitive to the initialisation of the first latent code z_0. Abdal et al. 2019b show that different initialisation leads to different perceptual results. They found that using the average latent code as initialisation achieves the best performance. Figure 2.8 shows the results of inverting images into the latent space of a pre-trained StyleGAN image generator using the optimisation-based method in comparison to other encoder-based methods.

2.5.3 Hybrid-based GAN inversion

Hybrid-based GAN inversion methods [J.-Y. Zhu et al. 2018; Bau et al. 2019; Roich et al. 2021; Alaluf et al. 2022] combine the strengths of encoder-based and optimization-based approaches to overcome the limitations inherent in each method. They first utilize a learning-based encoder to predict an initial latent code approximation and then refine this code using an optimization algorithm. The encoder's role is to speed up the optimization process by providing a suitable initial latent code. In J.-Y. Zhu et al. 2018, an initial latent code $\bar{z}_0$ for a given image x is

Figure 2.8: Qualitative results of different GAN inversion methods. The first column represents the input image and the second column shows the output of the optimisation method introduced by Abdal et al. 2019b. The remaining two columns show the results of using encoder-based methods: pSp encoder [Richardson et al. 2021] and ReStyle encoder [Alaluf et al. 2021], respectively. Figure is adapted from Alaluf et al. 2021

predicted by training an encoder on a dataset of training images. This computed $\bar{z}_0$ is then used as the starting point for the optimization-based method to enhance the visual reconstruction quality. Roich et al. 2021 introduces a different technique to improve visual fidelity by using the computed $\bar{z}_0$ latent code as a pivot to further fine-tune the image generator, ensuring that the given image x is faithfully reconstructed. This technique is referred to as pivotal tuning.

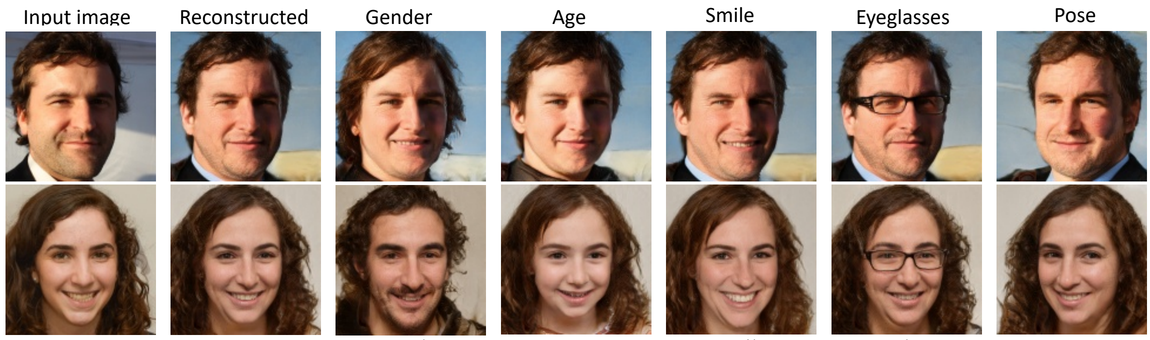
2.5.4 Pros and Cons of GAN Inversion Methods

Encoder-based methods predict the latent code of a given image in less than a millisecond during inference on a modern GPU (e.g. NVIDIA V100 GPU with 32 GB of memory), as they employ a feed-forward network. This makes them suitable for real-time applications, such as audio-driven talking-head generation. However, these methods require training an encoder network on a dataset, which may limit their generalization capabilities, and the visual quality of the reconstructed image might still require improvement. On the other hand, optimization-based methods produce very accurate reconstructions, which are important for talking-head video generation but are significantly slower compared to encoder-based methods. Hybrid-based methods combine the strengths of both approaches: they are faster than optimization-based methods by computing the starting point using the encoder-based method, and they produce better reconstructions than the encoder-based method alone by further refining the initial result with the optimization method. Nevertheless, hybrid-based methods still inherit some of the limitations of each individual approach Xia et al. 2022.

2.6 Latent space understanding

Some works have studied the intermediate latent space of StyleGAN [Roich et al. 2021; Abdal et al. 2019a; Y. Shen et al. 2020; Härkönen et al. 2020] and discovered meaningful semantics for manipulating images. The goal is to find disentangled directions in the latent space of a pre-trained StyleGAN image generator that aligns with semantic attributes in the generated images. For example, one direction could be for age, which allows one to change the age of a generated from young to old or old to young.

Y. Shen et al. 2020 propose *InterFaceGAN*, a supervised method to find meaningful directions by training an attribute classifier in the intermediate latent space W. The method begins by generating image samples from the pre-trained StyleGAN model. Next, these synthesised



(a) Results of changing the semantics of given images using the *InterFaceGAN* method [Y. Shen et al. 2020]

(b) Results of changing the semantics of given images using two unsupervised methods: Härkönen et al. 2020 and Y. Shen and B. Zhou 2020

Figure 2.9: The figure shows the results of various methods examining the latent space of the StyleGAN for image manipulation.

images are annotated using a pre-trained facial attribute network, which predicts the attribute label, e.g., male or female. These annotations are then used to train a linear Support Vector Machine (SVM) for each attribute to find a hyperplane in the latent space that separates two labels of the same attribute. To change a specific attribute of a given image, the method finds its latent code using GAN inversion methods and moves it from one side of the attribute's hyperplane to the other side. Figure 2.9a shows the results of the *InterFaceGAN* method.

Other works [Voynov and Babenko 2020; B. Wang and Ponce 2021; Härkönen et al. 2020; Y. Shen and B. Zhou 2020] find meaningful directions in an unsupervised form. Härkönen et al. 2020 apply Principal Components Analysis (PCA) in the latent space of a pre-trained StyleGAN model to identify important latent directions. They demonstrate that meaningful directions can be found by layer-wise perturbation along the principal directions. Similarly, Y. Shen and B. Zhou 2020 find interpretable directions in the latent space by performing eigenvector decomposition on the generator's weights. Figure 2.9b demonstrates the results of these two methods.

2.7 Unconditional video generation

Various approaches [Yushchenko et al. 2019; Vondrick et al. 2016; Saito et al. 2017; Saito et al. 2020; Acharya et al. 2018; Tulyakov et al. 2017; Tian et al. 2021; Skorokhodov et al. 2022] have been proposed for the task of unconditional video generation. Vondrick et al. 2016 propose a GAN-based approach VGAN to disentangle content and motion by training two parallel generators: foreground and background generators. The foreground generator models the motion part of a video using a 3D convolutional neural network while the background generator models the content part using a 2D convolutional neural network. Both generators take a single noise vector as input and their outputs are combined to generate a video. Saito et al. 2017 propose TGAN, a different approach by training a temporal generator and frame generator. The temporal generator outputs a sequence of latent vectors that correspond to a sequence of video frames using a single noise vector as input. The frame generator then takes each latent vector as input and generates a video frame. Both VGAN and TGAN generate a video from a single noise vector. Tulyakov et al. 2017 propose an alternative approach by using two different noise vectors as input to a single generator: a content vector and a sequence of temporal vectors. The content noise vector is fixed across all frames and sampled from a Gaussian distribution to set the content of the generated video. The temporal noise vector is recurrently learned using an RNN-based network and is variant across each frame to set the dynamics of the generated video.

Previous approaches successfully attempt to generate videos; however, their generated video is with low-resolution. Generating high-resolution videos requires high computational resources and increases the complexity of the model. Recently, there have been several works [Tian et al. 2021; Fox et al. 2021] that generate high-resolution videos by leveraging a pre-trained StyleGAN image generator [Karras et al. 2019]. Tian et al. 2021 propose a MoCoGAN-HD model that uses a motion generator to predict residual latent codes from an initial latent code sampled from the latent space of a pre-trained StyleGAN generator. The model is trained in the image space with a multi-scale video discriminator as well as a contrastive image discriminator. Similarly, Fox et al. 2021 reduce the training cost by training a Wasserstein GAN model in the latent space instead of image space. Although their model is trained on a single subject dataset, it can transfer the learned motion to a new subject using an offset trick. Skorokhodov et al. 2021 modify the StyleGAN network to learn a continuous latent trajectory using a neural representation-based approach. Video samples generated using the aforementioned models can be seen in Figure 2.10.

2.8 Conclusion

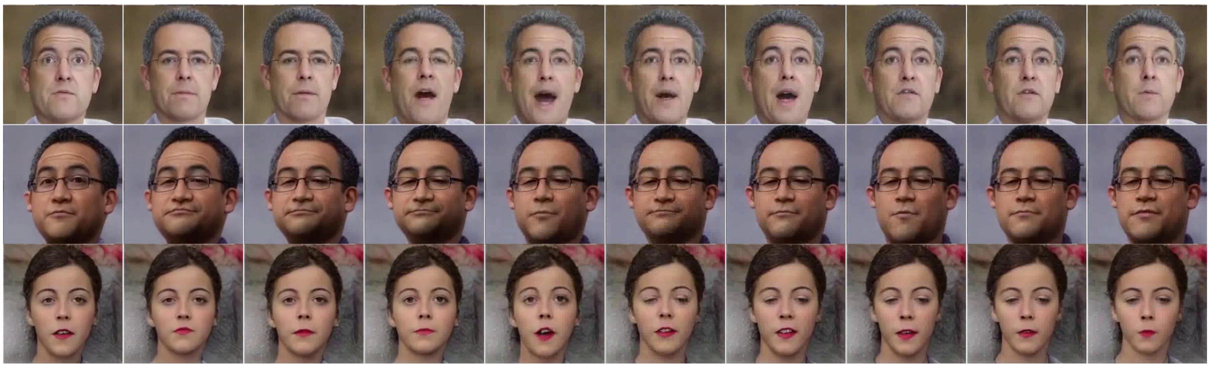
In this chapter, we provide an overview of methods for synthesising talking-head videos, including early works that use HMM models and sample-based methods. We also cover methods that employ deep learning to synthesise talking-head videos, including recent ones that utilise transformer models or diffusion-based models. Additionally, we describe commonly used datasets for talking-head videos. Furthermore, we discuss advancements in image synthesis over the past decade, with a specific focus on the StyleGAN model. This is followed by an explanation of GAN inversion methods: optimisation-based, encoder-based, and hybrid-based methods. We present the results of these methods, which aim to understand the latent space of a pre-trained StyleGAN model in order to identify semantic directions. The GAN inversion methods presented in this chapter are limited to projecting single images into the latent space of a pre-trained image generator.

In the next chapter, we study the optimisation-based method [Section 3.3.1] and the encoder-based method [Section 3.3.2] for projecting video frames into the latent space of a pre-trained StyleGAN image generator. We demonstrate that the encoder-based approach can faithfully project video frames and can be integrated into a video generation model. We employ an image encoder in the video generation model to generate videos from random vectors (Section 3.4).

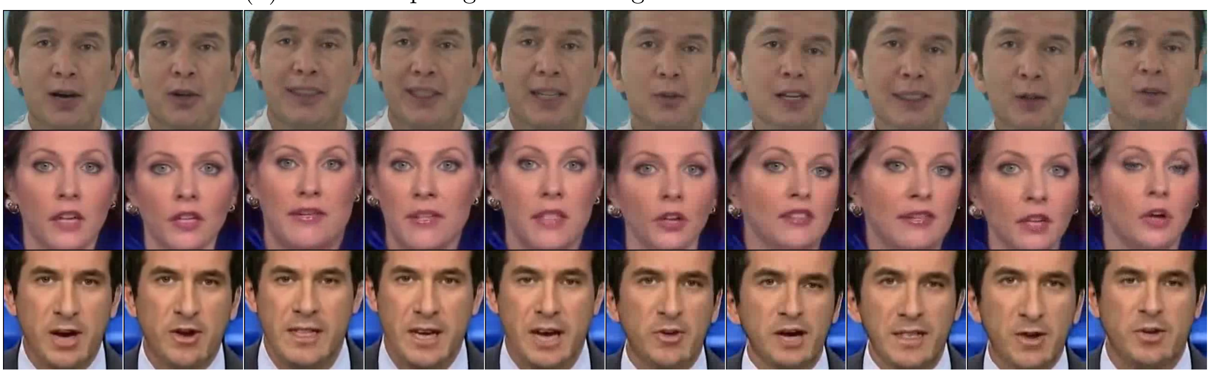
Lastly, in this chapter, we discuss unconditional video generation methods, particularly those built on top of the StyleGAN model. These methods demonstrate that the StyleGAN image generator can be used to synthesise videos; however, the StyleGAN model has not been used for synthesising audio-driven talking-head videos. In Chapter 4, we propose a novel approach that uses the StyleGAN model to synthesise talking-head videos from speech audio.

The talking-head video generation model presented in the next two chapters is built on the StyleGAN model. Although there are various GAN models, VAEs, and diffusion models that generate images from noise vectors or texts, we prefer StyleGAN for several reasons. Firstly, StyleGAN's intermediate latent space W is shown to be disentangled and semantically meaningful [Section 2.6], unlike the entangled latent space of ProgressiveGAN [Karras et al. 2019]. This

(a) Video samples generated using MoCoGAN-HD [Tian et al. 2021].



(b) Video samples generated using the work of Fox et al. 2021



(c) Video samples generated using the work of Skorokhodov et al. 2021

Figure 2.10: The figure shows video samples generated using three different video generative models built on top of the StyleGAN model.

makes StyleGAN more suitable for video generation, as discussed in Section 2.7. Additionally, the computational efficiency of StyleGAN in our talking-head system as a feedforward network is significantly higher compared to the slow training and inference times of diffusion-based image generators. Moreover, StyleGAN can generate high-fidelity images, surpassing VAE-based generators known for producing blurrier images [Goodfellow 2017].

The StyleGAN image generator can synthesize high-resolution images, which enables our model to synthesize high-resolution videos (Section 4.3). Most existing talking-head methods generate videos that are typically limited in terms of overall realism and resolution. In addition, using a pre-trained StyleGAN image generator in our system reduces the computational cost of training a talking-head video model that synthesizes high-resolution videos, as the number of learnable parameters would otherwise be very large.

Chapter 3

Video generation using a learned image encoder

3.1 Introduction

In this chapter, we train an image encoder to project videos into the latent space of the StyleGAN image generator and propose an unconditional video generation method that utilises the learned image encoder to generate videos. We begin by exploring two projection methods: encoder-based and optimisation-based. We learn to project a talking-head video into the latent space frame by frame, resulting in a latent trajectory (a sequence of latent codes), which is then used to reconstruct the video. We conduct experiments with the StyleGAN image generator pre-trained on a dataset of static images of human faces (FFHQ dataset) and a dataset of talking-head videos. We demonstrate that projecting a video into the latent space and regenerating the video from the projection results in a temporally coherent reconstruction.

Using the learned encoder-based projection, we develop a GAN-based model that generates latent trajectories in the latent space of a pre-trained StyleGAN image generator, corresponding to realistic talking-head videos. This is different from previous work because we use an image encoder to preserve the identity of the subject in the video and to make the training process more effective. The model is trained to generate a talking-head video by conditioning it on noise samples. The trajectories originate from the latent vector corresponding to a given identity image, obtained by projection using an image encoder. A family of trajectories is generated by conditioning the generator on a random sample from a normal distribution. We experiment with the StyleGAN image generator pre-trained on video frames of talking heads, as well as pre-trained on the FFHQ dataset of stationary heads, i.e., not talking. We train the model using two commonly used talking-head datasets: TCD-TIMIT and GRID. The results demonstrate that we can generate plausible latent trajectories that produce realistic talking-head videos similar to those in the training datasets. We examine the results and show that the perceived identities remain consistent and the generated trajectories remain within the subspace of plausible head appearances for the given identity. Furthermore, we show that the model learns to disentangle motion from appearance without being explicitly trained to do so.

3.2 Datasets

In this section, we describe two datasets of talking-head videos used in this chapter and in the subsequent two chapters: TCD-TIMIT and GRID.

TCD-TIMIT Dataset. The TCD-TIMIT dataset [Harte and Gillen 2015] is notable for its comprehensive collection of visual and auditory speech data. It includes recordings from approximately 59 volunteers, comprising 32 male and 27 female speakers with a variety of accents, primarily Irish. Each speaker in the dataset recites about 98 sentences, sourced from the TIMIT corpus, resulting in nearly 6,800 unique sentences in total. An example sentence is“She had your dark suit in greasy wash water all year”. The videos are of high quality, typically recorded at a resolution of 1920x1080 pixels, ensuring clear visibility of the speakers' facial movements and lip articulations. The audio is recorded in a noise-free environment, with a high sampling rate to capture the nuances of speech. The videos have a frame rate of 30 frames per second.

GRID Dataset. The GRID dataset [Cooke et al. 2006] is a more structured dataset, containing recordings from 34 speakers (18 male and 16 female). Each speaker delivers 1,000 sentences, leading to a total of 34,000 sentences across the entire dataset. The sentences are constructed using a fixed, six-word format that includes combinations of specific command words, colours, prepositions, letters, numbers, and adverbs, such as “Place blue at A1 now.” The video recordings are of standard quality, with clear visibility for lip-reading purposes at a resolution of 720x576 pixels. The audio component is also recorded in a controlled, noise-free environment. The videos have a frame rate of 25 frames per second. The GRID dataset includes speakers from a more diverse range of ethnicities and skin colours. Figure 2.2 shows examples of both datasets.

3.3 GAN inversion for video projection

The aim of GAN inversion is to find the latent code of a given input image in the latent space of a pre-trained GAN generator. The majority of GAN inversion methods are to project a single image into the latent space of a pre-trained StyleGAN image generator. However, there is limited work that studies GAN inversion methods for projecting a video i.e. a sequence of frames into a latent space of a pre-trained image generator [Oorloff and Yacoob 2022]. Therefore, in the following, we analyze two GAN inversion methods for projecting videos: optimisation-based and encoder-based methods. This would provide insight into the latent space for video generation.

3.3.1 Optimisation-based GAN inversion

The optimisation-based method for inversion finds an optimised latent code that generates a similar image to a given image under a distance function. This method embeds a single image at a time by starting from an initialised latent code and then optimises it directly by computing a loss between the output image and the original image. For StyleGAN, there are three latent spaces into which we can embed an image. Figure 3.1 shows the three latent spaces of StyleGAN: Z, W and W^+ spaces. The initial latent space Z is a normal distribution. The random noise vectors $z \in Z$ are converted to an intermediate latent space W of 512 dimensions via eight fully connected layers. The extended latent space W^+ is a concatenation of 18 distinct 512-dimensional w vectors, each corresponding to a layer of the StyleGAN generator. The intermediate latent space W is considered to be more disentangled than Z [Karras et al. 2019], making it a better choice for image projection. Additionally, Abdal et al. 2019b demonstrate that using the extended intermediate latent space W^+ improves the generalisation of the projection method.

3.3.1.1 StyleGAN image generator pre-trained on FFHQ dataset

We analyse W and W^+ spaces of StyleGAN for video projection. The StyleGAN model is trained on the Flickr-Faces-HQ (FFHQ) dataset [Karras et al. 2019], which contains around

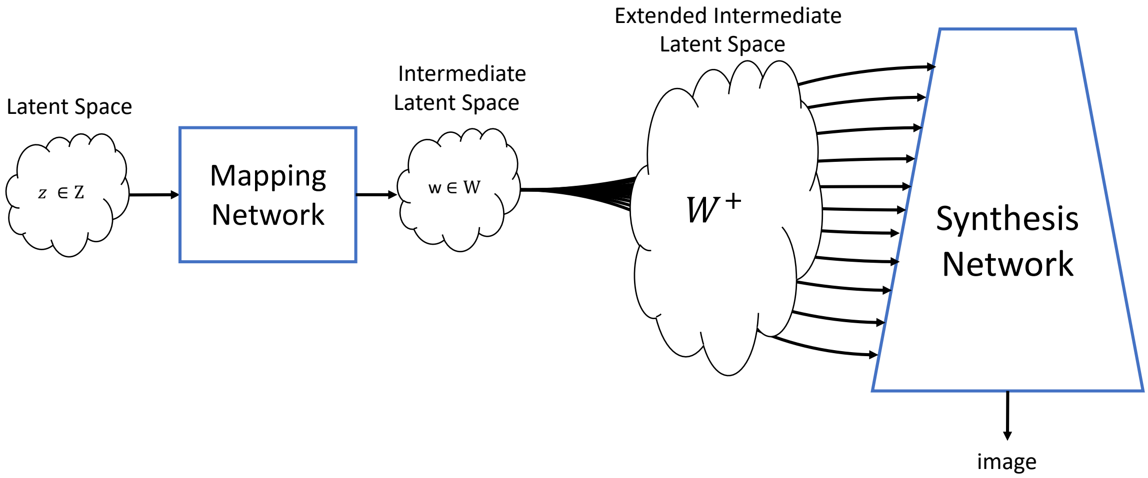


Figure 3.1: Three latent spaces of the StyleGAN model [Karras et al. 2019.]

70k high-resolution images of stationary heads (i.e., not talking) sourced from Flickr[1]. The StyleGAN model generates high-resolution images of size 1024x1024. Samples of the training images are shown in Figure 3.2, while some samples of the model's generated images are shown in Figure 3.3. To project a video into the latent space of StyleGAN, we follow the GAN inversion method of Abdal et al. 2019b for a single image. However, we modified it for a sequence of frames rather than a single image as shown in Algorithm 1. The w_i^* is initialised using the average latent code $\bar{w}$ computed by sampling random vectors z and passing them through the mapping network to get w vectors. To measure the loss between the embedded video and the original input video, we use L_2 loss and a perceptual loss i.e. L_{LPIPS} [R. Zhang et al. 2018].

Dataset pre-processing. For video projection, we use the TCD-TIMIT dataset of talking-head videos [Harte and Gillen 2015]. The TCD-TIMIT dataset comprises 59 speakers each uttering 100 sentences. The framerate of the original videos is 30 fps. We re-sampled the videos to be 25 fps using the FFmpeg [2] software tool which employs a frame-dropping technique. We track faces in the videos and extract their landmarks using the Face Alignment library [Bulat and Tzimiropoulos 2017]. We align and crop the video frames following the same method used in the FFHQ dataset [Karras et al. 2019]. Figure 3.4 shows an overview of pre-processing a single frame.

We then apply Algorithm 1 to project video frames into W and W^+ spaces separately. Figure 3.5 shows the result of projecting a video sample from the TCD-TIMIT [Harte and Gillen

[1] `https://www.flickr.com/`
[2] `https://ffmpeg.org/`

Figure 3.2: Sample images from the FFHQ dataset used for training StyleGAN model [Karras et al. 2019].

2015], where the second row represents the projection into the W space while the third row represents the projection into W^+ space. We can see that projecting into the extended latent space W^+ achieves a better reconstruction quality than projecting into the W latent space. The intermediate latent vectors w of W^+ space can be distinct from each other and each can be fed into different convolutional layers of the StyleGAN's generator. Therefore, the W^+ space has higher degrees of freedom than the W space. Thus, we use W^+ space for video projection as it is more expressive than the W space.

Figure 3.3: Sample images generated using the StyleGAN model trained on the FFHQ dataset [Karras et al. 2019].

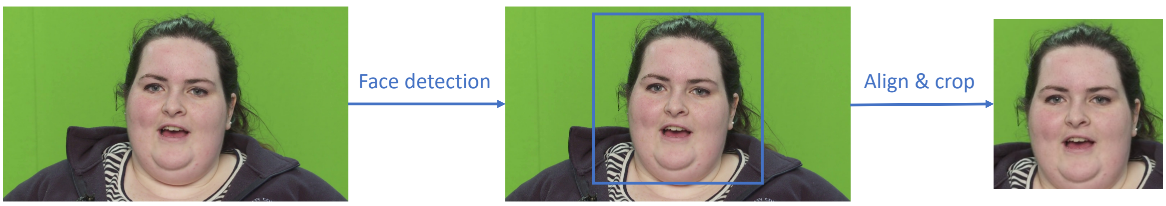


Figure 3.4: Overview of pre-processing a video frame from the TCD-TIMIT dataset. We employ the same methods used to pre-process the FFHQ dataset for training the StyleGAN model.

Algorithm 1: Video projection into the latent space of StyleGAN

Input: A video $\{x_t\}_{t=1}^{T}$, where T is the number of frames and $x_t \in \mathbb{R}^{w \times h \times 3}$ are the individual frames to project; a pre-trained image generator $G(\cdot)$.

Output: The embedded latent codes $\{w_t^*\}_{t=1}^{T}$ and the embedded video $\{G(w_t^*)\}_{t=1}^{T}$ optimised via F'.

Initialize $t = 1$;
while $t < T$ **do**
  Initialize latent code $w_t^* = \bar{w}$;
  while *not converged* **do**
    $L \leftarrow L_{LPIPS}(G(w_t^*), x_t) + \frac{\lambda}{T}\|G(w_t^*) - x_t\|_2^2$;
    $w_t^* \leftarrow w_t^* - \eta F'(\nabla_{w_t^*} L\ )$;
  end
  $t = t + 1$;
end

Algorithm 1 is very slow as the number of optimisation steps required for convergence is typically around 5000 and it takes on average 6 minutes on NVIDIA Tesla V100 with 32 GB of memory size. Figure 3.6 shows a plot of the loss values as the number of the optimisation steps increases on projecting into both spaces. From the figure, it can be seen the losses of both converge near 1000 steps. The Algorithm projects each frame independently without taking into account that consecutive frames need to be close in the latent space. Moreover, projecting the same video again can lead to two different projections. We can see this by visualising the projection using the first two components of the PCA computed for each video. Figure 3.7a shows the 1st projection attempt of a video while 3.7b shows the 2nd projection attempt of the same video with the same number of optimisation steps. This is as result of the non-convexity of the latent space as well as the stochasticity of the optimisation Algorithm.

To overcome the aforementioned challenges, we propose an alternative Algorithm to project a video into the latent space. We initialise the latent code w_t^* of frame x_t using the optimised latent code w_{t-1} of the previous frame x_{t-1}. For the first frame x_1, we initialise its latent code w_1^* from the average latent code $\bar{w}$. This is illustrated in Algorithm 2. We apply this Algorithm to project the same video and show the visual results in comparison to Algorithm 1 in Figure

Figure 3.5: A comparison between a video projection using the optimization-based method into the W latent space (middle row) and the W^+ latent space (bottom row). The real video (first row) is from the TCD-TIMIT dataset.

3.9. The results of the two algorithms appear to be very similar in the figure. To speed up the projection, we set the number of the optimisation steps to 500 except for the first frame, set to 5000 steps. We found that 500 iterations were sufficient for convergence. Figure 3.8 visualises the projection results of using both Algorithms in the latent space based on PCA computed per video. We can see Algorithm 2 projects a video as a smooth trajectory in the latent space (Figure 3.8b) while Algorithm 1 scatters the projection of a video (Figure 3.8a).

Although the visual results of both algorithms are very similar (as illustrated in Figure 3.9), Algorithm 2 produces smooth linear projection results. This is due to the initialization of the latent code w_t^* for frame x_t using the optimized latent code w_{t-1} from the previous frame x_{t-1}, as shown in lines 3 to 7 of Algorithm 2. In addition, Algorithm 2 is faster than Algorithm 1 because the number of optimization steps for projecting frames after the first frame is 500 compared to 5000 in Algorithm 1. More results comparing both Algorithms are presented in Appendix A.

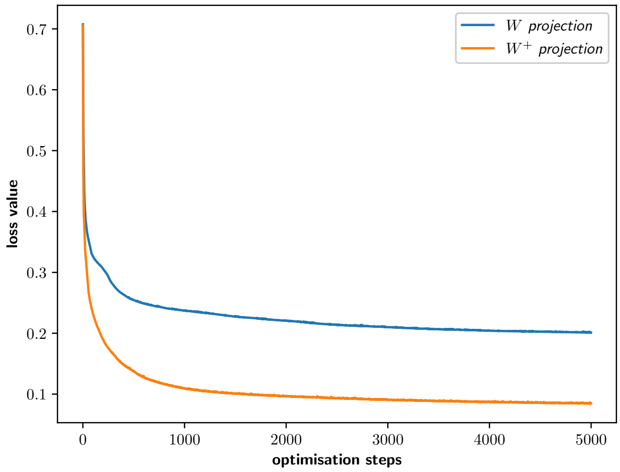
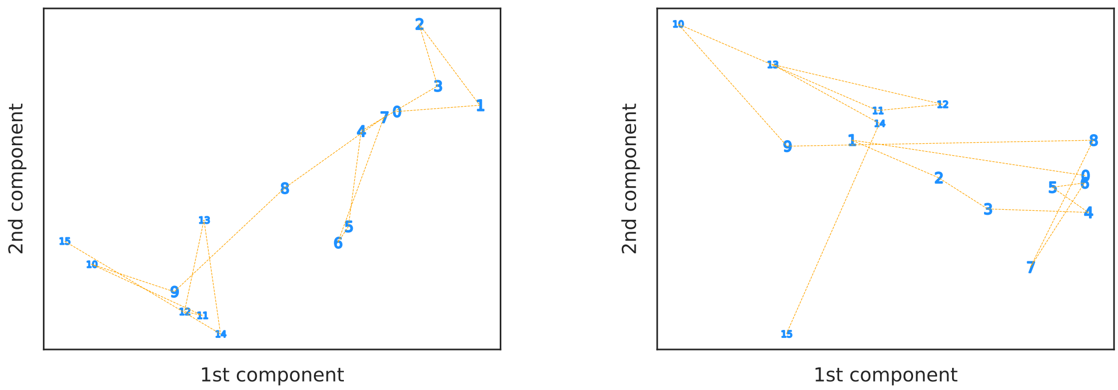
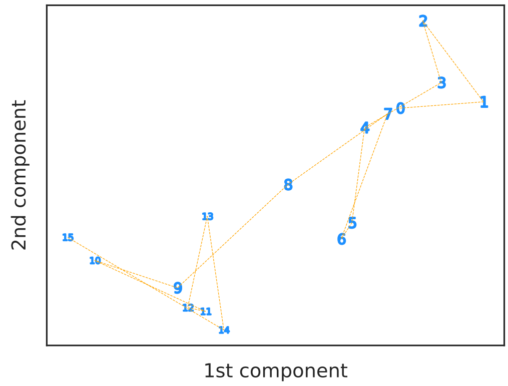


Figure 3.6: Comparison of the loss values during the optimization process when projecting a video frame into W and W+ spaces using an optimisation-based approach. The video frame is taken from the TCD-TIMIT dataset.

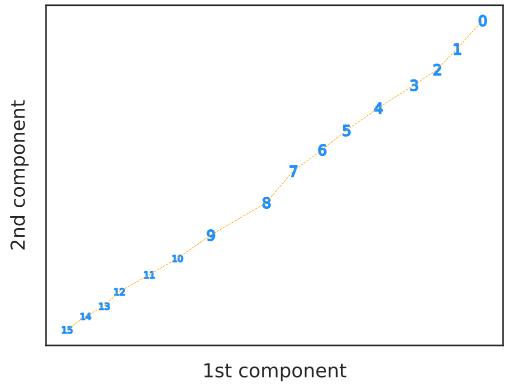


(a) 1st projection (b) 2nd projection

Figure 3.7: Visualisation of two distinct projection attempts of the same video using the first two principal components computed per video. It is evident that the video projection using Algorithm 1 of the optimization-based method resulted in two different projections. The video sample comes from the TCD-TIMIT dataset.



(a) Projection using Algorithm 1



(b) Projection using Algorithm 2

Figure 3.8: Visualisation of the video projection results using Algorithm 1 and Algorithm 2, based on the first two principal components computed for each video. Algorithm 2 projects a video as a smooth trajectory within the latent space. The video sample is taken from the TCD-TIMIT dataset.

Algorithm 2: Video projection into the latent space of StyleGAN

Input: A video $\{x_t\}_{t=1}^{T}$, where T is the number of frames and $x_t \in \mathbb{R}^{w \times h \times 3}$ to embed; a pre-trained image generator $G(\cdot)$.

Output: The embedded latent codes $\{w_t^*\}_{t=1}^{T}$ and the embedded video $\{G(w_t^*)\}_{t=1}^{T}$ optimzed via F'.

Initialize $t = 1$;
while $t < T$ **do**
  if $t = 1$ **then**
    Initialize latent code $w_t^* = \bar{w}$;
  else
    Initialize latent code $w_t^* = G(w_{t-1}^*)$;
  end
  while *not converged* **do**
    $L \leftarrow L_{LPIPS}(G(w_t^*), x_t) + \frac{\lambda}{T}\|G(w_t^*) - x_t\|_2^2$;
    $w_t^* \leftarrow w_t^* - \eta F'(\nabla_{w_t^*} L\ )$;
  end
  $t = t + 1$;
end

Figure 3.9: The middle row shows video frames projected into the W^+ latent space with the use of Algorithm 1, while the bottom row displays the results of using Algorithm 2. The original video frames, shown in the first row, come from the TCD-TIMIT dataset. Visually, it can be seen that the quality of the video frames projected using Algorithm 2 is comparable to those produced by Algorithm 1.

3.3.1.2 StyleGAN image generator pre-trained on video frames of a talking head

Our previous experiments are on a StyleGAN generator trained on the FFHQ dataset [Karras et al. 2019]. The latent space is richly diverse and disentangled into prominent features such as gender, age, hair, etc. However, other features that are prominent in video datasets such as mouth movements are less represented in the latent space. In this section, we experiment with the same StyleGAN image generator being pre-trained on video frames of talking heads rather than images of stationary heads.

Training StyleGAN on video frames. We start by training the StyleGAN model on video frames of TCD-TIMIT with 48 speakers. The StyleGAN model is the same as the one we discuss in Section 3.3.1.1 but we train it to generate images of size 256x256. Samples of training data are shown in Figure 3.10. To facilitate the training, we fine-tune the pre-trained StyleGAN of FFHQ further using the same training strategy used in training the original model. Starting from a pre-trained model of face generation eases the convergence of the model since the model already knows how to map from a normal distribution to an image domain of human faces. We measure the performance of the training using the Fréchet Inception Distance (FID) [Heusel et al. 2018] as the most common metric for image generation tasks. The FID measures the similarity between the distribution of real images and generated images based on feature representations extracted from a pre-trained neural network. Figure 3.11 shows generated samples of the StyleGAN generator pre-trained on video frames. Since the training data contains only

Figure 3.10: Sample images of speakers from the TCD-TIMIT talking-head video dataset [Harte and Gillen 2015]. This dataset is used to train the StyleGAN model.

48 speakers, the StyleGAN model is not able to generate novel faces outside these identities as the original model. We can see in Figure 3.11 that the model learns to generate the appearance of speakers from the training set but with different mouth shapes and expressions.

Latent space interpolation. We further analyse the latent spaces Z and W by performing linear interpolation and spherical interpolation [Shoemake 1985]. It is essential for the latent space to be continuous and disentangled, especially for video generation. Figure 3.12 shows the two interpolation methods between two latent codes in both latent spaces, Z and W. We can observe that interpolating in the W latent space results in visually better images with no artefacts, This is in comparison to those produced in the Z latent space, as highlighted in yellow in the figure. Although spherical interpolation is known to produce sharper images when visualising the latent spaces of generative models [White 2016], the distinction between this method and linear interpolation is explicitly illustrated in Figure 3.12.

Figure 3.11: Image samples generated using the StyleGAN generator pre-trained on video frames from the TCD-TIMIT talking-head dataset.

In the previous section, we projected video frames into the latent space of the StyleGAN image generator pre-trained on images of stationary heads i.e the FFHQ dataset using Algorithm 2. For this experiment, we project a video sample from one of the training speakers into the latent space W^+ of the StyleGAN image generator pre-trained on video frames of a talking head from TCD-TIMIT dataset using Algorithm 2. Figure 3.13 shows the projected video frames with the real ones. We can see that the projection method reconstructs the input frames very effectively. Since the generator has seen the speaker, the projection method converges quicker than the projection into the StyleGAN image generator pre-trained on the FFHQ dataset. This can be seen when we plot the loss values over the optimisation steps of the first frame in Figure 3.14. We also visualise the projection in the latent space in Figure 3.15.

Figure 3.12: Each row displays the results of interpolating between two latent vectors in the latent space of the StyleGAN model, pre-trained on the TCD-TIMIT dataset. The first two rows demonstrate the results of interpolation performed in the Z space, while the latter two rows present the results of interpolation in the W space. *lerp* denotes linear interpolation, whereas *slerp* denotes spherical interpolation. It is clear that interpolation in the W space yields more visually appealing video frames, as highlighted in yellow.

Figure 3.13: The results of projecting video frames into the latent space of StyleGAN, pre-trained on the TCD-TIMIT dataset Harte and Gillen 2015, using Algorithm 2 of the optimisation-based method.

3.3.2 Encoder-based GAN inversion

In the previous section, we discuss the projection of a video using an optimisation-based method. In this section, we study an encoder-based method to project a video into the latent space W^+. The idea is to use an encoder feed-forward network to invert the image generator. For the StyleGAN model, the encoder is trained to project a given image into the latent space W^+ of the image generator. In the following, we start by experimenting with the StyleGAN model pre-trained on the FFHQ dataset of faces with static expressions (not talking) [Karras et al. 2019]. Second, we experiment with the StyleGAN model pre-trained on video frames of talking heads. We show the results of projecting videos in both experiments using an encoder-based method.

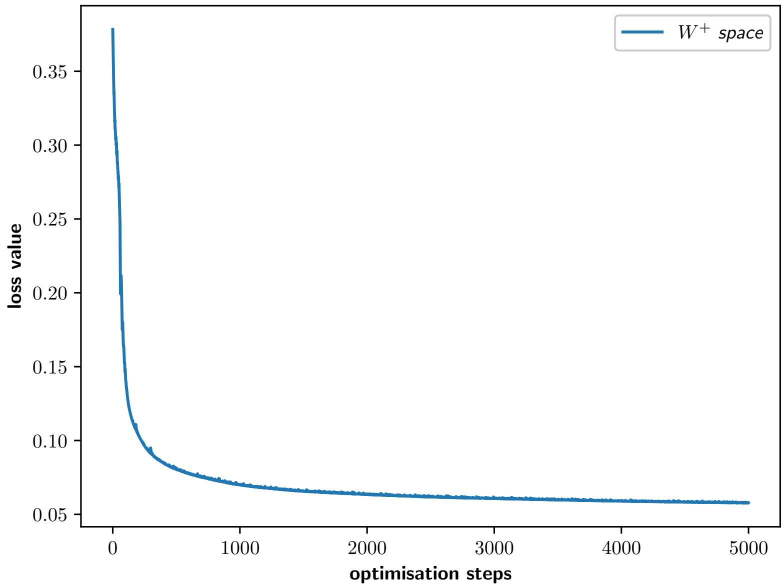


Figure 3.14: The loss plot for projecting a single video frame from the TCD-TIMIT dataset into the latent space of the StyleGAN generator, which was pre-trained on talking-head video frames from the same dataset, using the optimisation-based method described in Algorithm 2.

3.3.2.1 StyleGAN image generator pre-trained on FFHQ dataset

For StyleGAN pre-trained on the FFHQ dataset, Richardson et al. 2021 propose the pSp encoder model to project any real image into the latent space. The pSp encoder is trained with the same training dataset, FFHQ. Figure 3.16 shows the encoder framework. The architecture choice was motivated by the latent code layers being divided into three levels: coarse, middle and fine. The coarse-level layers correspond to the low-resolution convolutional layers (4^2 - 8^2), the middle-level layers to (16^2 - 32^2) and the fine-level layers correspond to (64^2 - 1024^2). The objective is to extract three levels of features from a backbone model using the pyramid feature technique [Lin et al. 2017]. This technique leverages the hierarchical structure of CNNs to simultaneously extract features from multiple layers, capturing high-level abstractions and fine-grained details. These extracted features are used to compute the corresponding latent codes by using a separate fully-convolutional model, *map2style*, for each computed latent code [Richardson et al. 2021].

The model was trained with common reconstruction losses: a pixel-wise L_2 loss and a feature loss. To preserve the identity of the input image, the model was trained on an additional loss to maximise the cosine similarity between the input and the output image in the feature space of the pre-trained ArcFace model [J. Deng et al. 2021]. Moreover, the model was encouraged

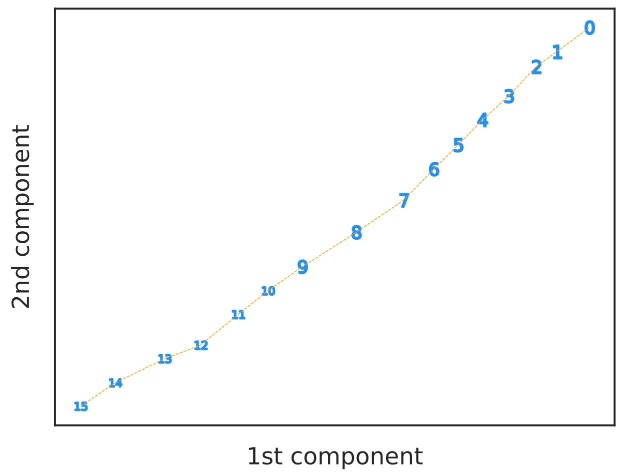


Figure 3.15: Visualisation of the video projection results using Algorithm 2 into the latent space of the StyleGAN generator pre-trained on video frames of talking heads from the TCD-TIMIT dataset. The visualisation is based on the first two principal components computed from the latent vectors obtained through the projection.

to project images close to the average latent code with a regularisation loss.

We use the pre-trained pSp encoder to project video frames from the TCD-TIMIT dataset into the latent space W^+ of the StyleGAN image generator. Figure 3.17 shows video frames projected into the latent space. We can see that the reconstructed frames are close to the real ones. The model was able to consistently preserve the identity within the frames. Interestingly, the model was able to accurately reconstruct the facial expressions e.g. mouth movements even though the model was trained on static images of people posing. However, we can see that the reconstructed images observe some distortion because neither the encoder nor the generator

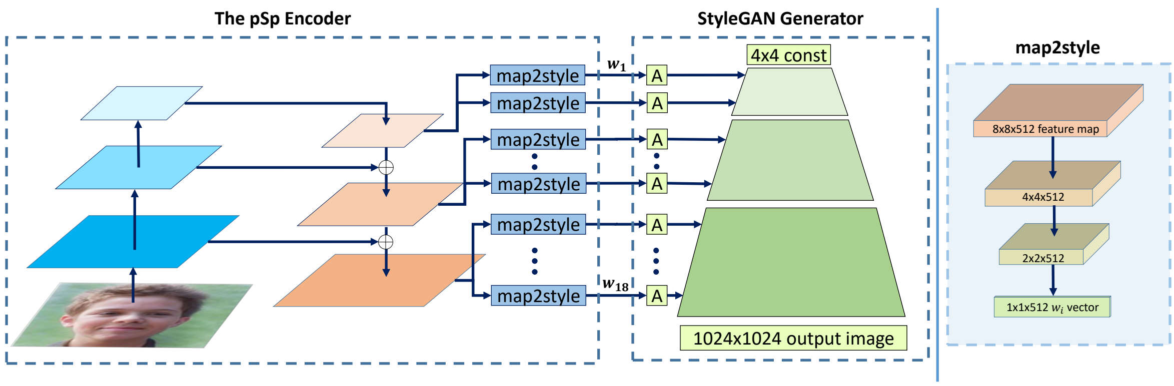


Figure 3.16: The pSp Encoder framework which we use to project video frames of talking heads. The *map2style* network is a fully convolutional network to gradually reduce the spatial size. Image is taken from Richardson et al. 2021

had seen these particular identities before. Figure 3.18 visualises the projection of the latent space. The encoder-based projection takes a fraction of a second to compute.

Figure 3.17: The results of projecting video frames from the TCD-TIMIT into the latent space of the StyleGAN model pre-trained on the FFHQ dataset using the pSp image encoder [Richardson et al. 2021].

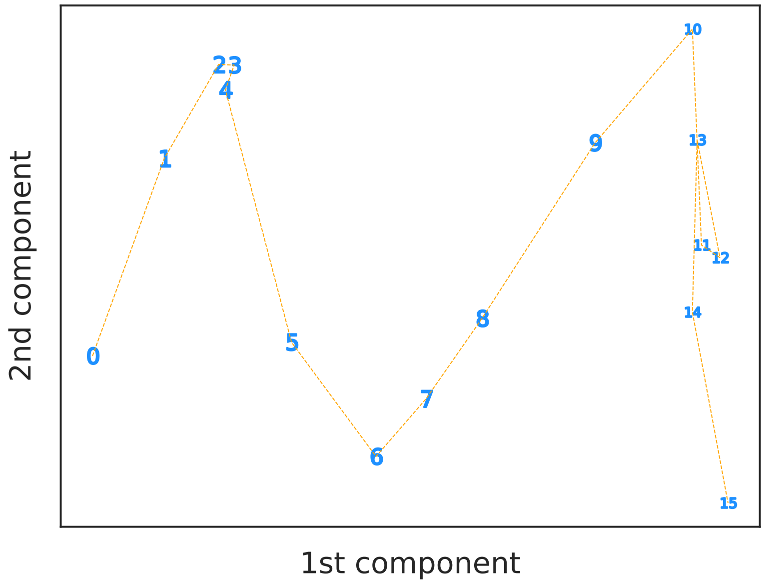


Figure 3.18: Visualisation of the video projection results using using the pSp image encoder [Richardson et al. 2021] into the latent space of the StyleGAN generator pre-trained on the FFHQ dataset. The visualisation is based on the first two principal components computed on the latent vectors obtained from the projection.

3.3.2.2 StyleGAN image generator pre-trained on video frames

We explore video frame projection into the latent space of a StyleGAN image generator pre-trained on video frames of talking heads rather than face images with static expressions i.e. the FFHQ dataset. For this experiment, we use the previous pre-trained StyleGAN model we discussed in Section 3.3.1.2. We choose the ResNet18 model [He et al. 2016] as our image

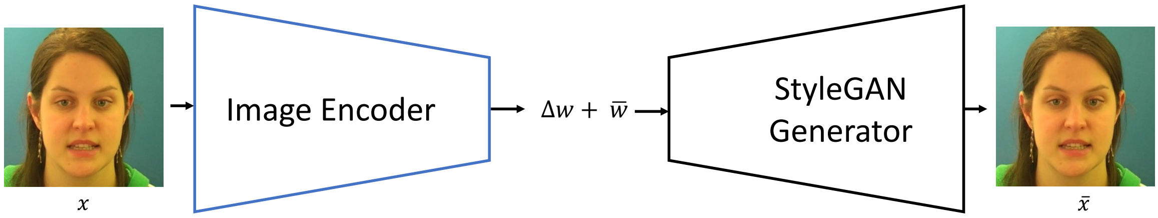


Figure 3.19: The framework for training an image encoder to project video frames into the latent space of the StyleGAN image generator.

encoder E_I and train it on video frames of a talking-head dataset. However, we modified the last layer's size to match the latent space size. Table 3.1 explains the details of the encoder network. We use the TCD-TIMIT dataset which is a talking-head video dataset and is the same training data used to pre-train the StyleGAN model. Figure 3.19 shows the training framework of the image encoder. The model predicts the offset vector to the average latent vector $\bar{w}$ to facilitate the training. The training is similar to the auto-encoder training framework where the StyleGAN generator is the decoder. However, the image encoder is only trained while the decoder is pre-trained and kept fixed. For the loss function, we use a pixel-wise L_2 loss defined as follows:

$$\mathcal{L}_{\mathrm{L}_2} = ||x_t - G(\bar{w} + E_I(x_t))||_2 \tag{3.1}$$

where x_t is a real frame at time t. Additionally, we use a feature loss to improve the perceptual similarities between the two frames:

$$\mathcal{L}_{\mathrm{LPIPS}} = ||\phi(x_t) - \phi(G(\bar{w} + E_I(x_t)))||_2 \tag{3.2}$$

where ϕ is a pre-trained VGG [Simonyan and Zisserman 2015] network. The objective loss for training the encoder is defined as follows:

$$\mathcal{L}_{\mathrm{enc}} = \mathcal{L}_{\mathrm{LPIPS}} + \mathcal{L}_{\mathrm{L}_2} \tag{3.3}$$

Training the image encoder. We initialise the weights of the first 4 ResNet blocks of the encoder using the original weights of the ResNet model trained on the ImageNet dataset for image classification [He et al. 2016]. We train the remaining layers from scratch. We use the

Layer type	Kernel	Stride	Output	Activation
2d convolution	7x7	2	64x128x128	ReLU
2d max pooling	3x3	2	64x64x64	-
2d convolution	3x3	1	64x64x64	ReLU
2d convolution	3x3	1	64x64x64	ReLU
2d convolution	3x3	1	64x64x64	ReLU
2d convolution	3x3	1	64x64x64	ReLU
2d convolution	3x3	2	128x32x32	ReLU
2d convolution	3x3	1	128x32x32	ReLU
2d convolution	3x3	1	128x32x32	ReLU
2d convolution	3x3	1	128x32x32	ReLU
2d convolution	3x3	2	256x16x16	ReLU
2d convolution	3x3	1	256x16x16	ReLU
2d convolution	3x3	1	256x16x16	ReLU
2d convolution	3x3	1	256x16x16	ReLU
2d convolution	3x3	2	512x8x8	ReLU
2d convolution	3x3	1	512x8x8	ReLU
2d convolution	3x3	1	512x8x8	ReLU
2d convolution	3x3	1	512x8x8	ReLU
adaptive 2d average pooling	-	-	512x1x1	-
Linear	-	-	6144	-

Table 3.1: The table describes the image encoder architecture

Adam optimisation method for training the encoder with a learning rate of 0.004.

After training the encoder, we project video frames into the latent space of the StyleGAN image generator pre-trained on the video frames. Figure 3.20 shows samples of the projected video frames. We can see the encoder efficiently reconstruct the input frames with no distortions. This is due to the fact that both the encoder and the generator have seen the identity. We also visualise the projection in Figure 3.21. We investigate the robustness of the encoder by projecting video frames of unseen identities [Figure3.22]. Here we show samples of unseen identities from the TCD-TIMIT dataset. We can see the identity was not reconstructed successfully. However, the model kept the identity consistent within the frames with a good visual appearance. In addition, the model continuously reconstructed the facial expression with the frames.

Figure 3.20: The results of projecting video frames from the TCD-TIMIT dataset using an image encoder. The encoder is trained to map video frames of talking heads into the latent space of the StyleGAN image generator, which is pre-trained on the TCD-TIMIT dataset.

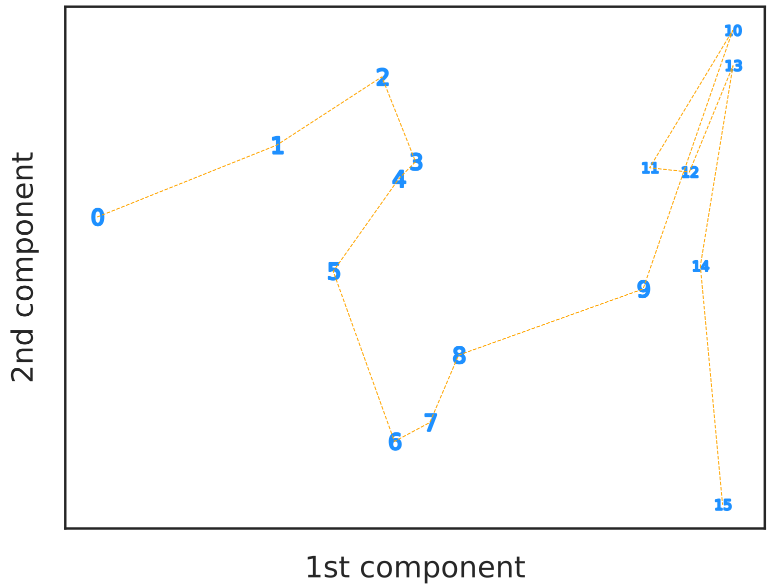


Figure 3.21: Visualisation of the video projection results using the image encoder into the latent space of the StyleGAN generator pre-trained on talking-head video frames. The visualisation is based on the first two principal components computed on the latent vectors obtained from the projection

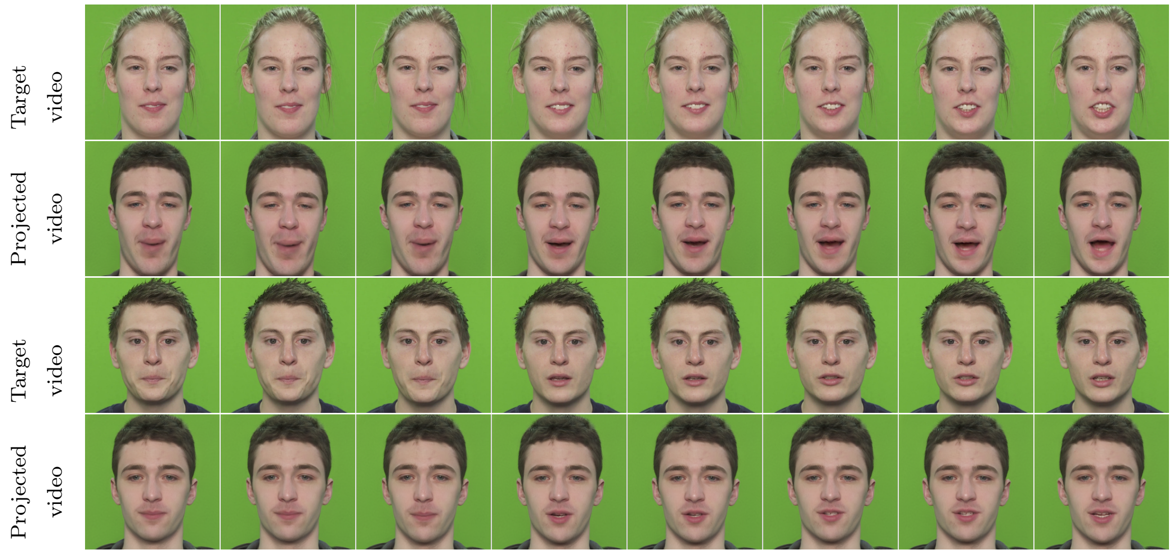


Figure 3.22: Results of projecting video frames from unseen speakers into the latent space of a StyleGAN pre-trained on the TCD-TIMIT dataset during the training of the image encoder. The image encoder struggles to faithfully reconstruct the target video frames; however, it maintains consistent identity representation across the projections.

3.4 Unconditional video generation

In this section, we explore the latent space of the pre-trained StyleGAN image generator for video generation. The task of video generation is similar to image generation using the learned encoder-based projection method. However, video generation is more challenging for various reasons [Tulyakov et al. 2017]. First, the task is a spatiotemporal generation meaning the model learns to generate plausible dynamics of object motion in addition to its visual appearance. Second, the additional time dimension adds extra variation. Consider the various speed at which a human performs a certain task e.g. walking. Third, human eyes are more sensitive to temporal discontinuity than visual artefacts in image generation.

Using a pre-trained image generator simplifies the video generation problem to focus on learning dynamics. The idea is to find a trajectory of latent codes in the latent space that corresponds to a sequence of frames in the image domain of the image generator. In this way, we decompose the video generation into appearance and motion. The model learns to start from a latent code that represents the initial appearance and then finds a sequence of deviations to the latent code which represents the motion.

Similar to the previous section on video projection, we experiment with the StyleGAN model pre-trained on video frames of talking heads as well as on faces with static expressions.

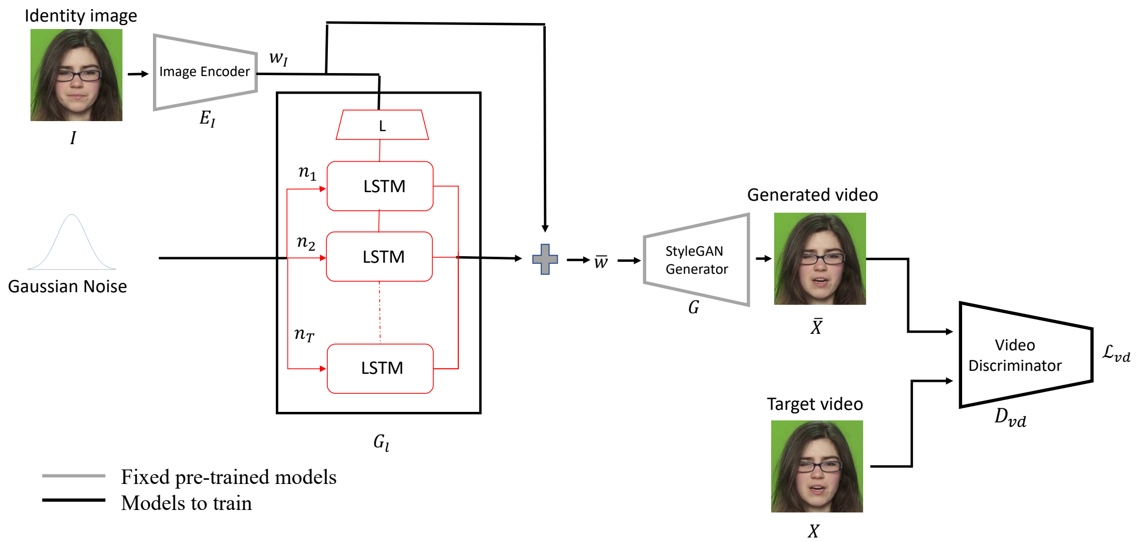


Figure 3.23: Overview of the unconditional video generation framework: Given an identity image I and a sequence of noise samples n_t, the aim is to generate a video of the given identity making typical facial expressions, such as in talking.

3.4.1 The approach

The proposed approach is based on a GAN and takes as input an identity image I to set the appearance of the video and a sequence of samples n_t from a normal distribution to set the dynamics of the video. First, the approach encodes an identity image I into the latent space of the pre-trained StyleGAN image generator to set the initial appearance for the generated video. For this, we use the learned encoder-based projection methods introduced in Section 3.3.2 resulting in a latent code vector w_I. Then, a recurrent function takes a sequence of samples n_t from a normal distribution to predict a sequence of displacements d_t that move the encoded latent code w_I into the latent space to represent the motion. The pre-trained StyleGAN image generator synthesises a video from this deviated sequence.

3.4.2 Framework

Our framework is inspired by the work of Tian et al. 2021. it has four components: an image encoder E_I, a latent generator G_l, a StyleGAN image generator G, and a video discriminator D_v. Both the image encoder and the StyleGAN image generator are pre-trained and fixed while training the latent generator and the video discriminator. Figure 3.23 gives an overview of the model.

The latent generator G_l has a linear layer L and an $LSTM$ unit. The linear layer provides the initial hidden inputs h_1 and c_1 of the $LSTM$ unit by taking as input the latent code w_I of the identity image. Given a sequence of noise samples $\{n_t\}_{t=1}^{T}$ where T the length of the generated video. the $LSTM$ outputs sequence of hidden vectors h_t to deviates the identity image latent code w_I.

$$h_t,\ c_t = LSTM(n_t, L(E_I(I))),\ where\ t = 1, 2, .., T \tag{3.4}$$

To ease the training, we apply the Principal Components Analysis (PCA) to the training video frames of the talking-head dataset encoded into the latent space of the pre-trained StyleGAN image generator. We select the first 384 principal components to form the matrix V, with these components as its columns, following the same choice as in the MoCoGAN-HD model [Tian et al. 2021]. The hidden outputs h_t generate the motion as follows:

$$\bar{w_t} = w_I + h_t * V \tag{3.5}$$

The pre-trained StyleGAN image generator generates a sequence of frames from $\bar{w_t}$ as follows:

$$x_t = G(\bar{w_t}) \tag{3.6}$$

Layer type	Kernel	Stride	Output	Activation
3d convolution	4x4	2	64x129x129	Leaky ReLU (0.2)
3d convolution	4x4	2	128x65x65	Leaky ReLU (0.2)
3d convolution	4x4	2	256x33x33	Leaky ReLU (0.2)
3d convolution	4x4	1	512x34x34	Leaky ReLU (0.2)
3d convolution	4x4	1	1x35x35	-

Table 3.2: The table describes the architecture of the video discriminator

3.4.3 Training

We train only the latent generator G_l and video discriminator D_v while we keep the image encoder E_I and the StyleGAN image generator G pre-trained and fixed. The latent generator is trained with the video discriminator using an adversarial loss. Given a real video X sampled from the training dataset and a fake video $\bar{X}$ generated using our framework, the goal of the video discriminator is to classify them as real or fake, respectively. However, the goal of the latent generator is to fool the discriminator into classifying fake videos into real ones. The video discriminator is a multi-scale 3D-convolutional network [Tian et al. 2021]. It is based on the image discriminator used in PatchGAN [Isola et al. 2017]; however, we condition each video frame on the given identity image to stabilize the training process. Table 3.2 shows the details of our video discriminator. Therefore, we train the parameters of the latent generator G_l and the video discriminator D_v using the following objective loss.

$$\min_{G_l} \max_{D_v} \mathcal{L}_{\mathcal{VD}}(D_v, G_l) = \mathbb{E}_{X \sim p_{\text{data}}}[\log D_v(X)] + \mathbb{E}_{I \sim p_{\text{data}}}[\log(1 - D_v(G(G_l(n, E_I(I)))))]. \quad (3.7)$$

where n is the noise samples drawn from a normal distribution.

To encourage the latent generator to generate diverse facial expressions and prevent it from falling into one expression, we train it with an information loss. The goal of the information loss is to maximize the mutual information between the given sequence of noise samples $\{n_t\}_{t=1}^{T}$ and the hidden outputs of the LSTM unit $\{h_t\}_{t=1}^{T}$. We first map the hidden outputs h_t using a two-linear layer network M and maximize the cosine similarity between the resulting vectors from the mapping network and the noise samples. Our information loss is defined as follows:

$$\max_{G_l} \mathcal{L}_{info}(G_l) = \frac{1}{T} \sum_{t=1}^{T} \text{CosineSim}(M(h_t), n_t), \quad (3.8)$$

where CosineSim is defined as

$$\text{CosineSim}(u, v) = \frac{u.v}{||u||\,||v||} \quad (3.9)$$

where u and v are vectors.

3.4.4 Implementation

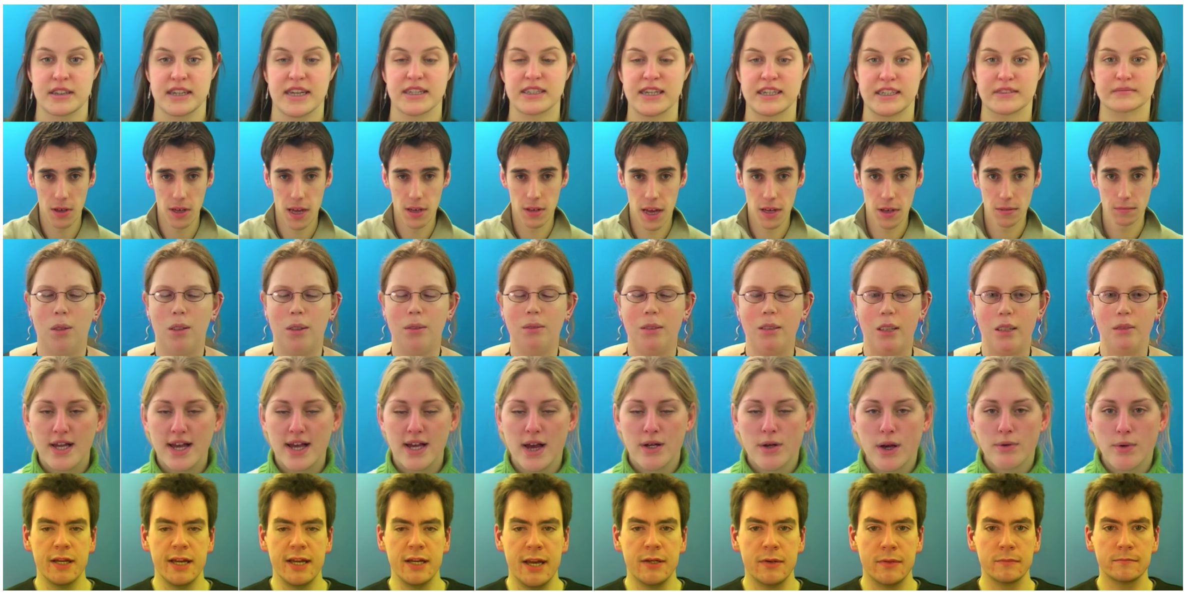
We use two talking-head video datasets: TCD-TIMIT [Harte and Gillen 2015] and GRID [Cooke et al. 2006]. The GRID dataset has 33 speakers uttering 1000 short sentences each containing 6 words while the TCD-TIMIT has 59 speakers each uttering 100 sentences. The video frames are pre-processed following the procedure described in Section 3.3.1.1. This is the same pre-processing method used in training the image encoder E_I (Section 3.3.2.2) and StyleGAN image generator G (Section 3.3.1.2). We implement our experiments using the deep learning framework PyTorch [Paszke et al. 2019] and run them on NVIDIA Tesla V100. For the image encoder E_I and the StyleGAN image generator G, we use the same pre-trained models described in the video projection Section 3.3.2.2. In our framework, we only train the latent generator G_l and the video discriminator D_v. We use the Adam optimiser [Kingma and Ba 2014] with a 0.0001 learning rate for both networks.

3.4.5 Results

We show the results of our framework where the image encoder and StyleGAN image generator used are pre-trained on the video frames of a talking-head dataset. The pre-trained image encoder is the same as the image encoder we introduce in Section 3.3.2.2 while the pre-trained StyleGAN image generator is introduced in section 3.3.1.2. Figure 3.24 shows video samples generated from noise samples using our framework trained on both datasets. The video samples have different photo-realistic facial expressions such as eye blinks and mouth movements. Moreover, we can see the identity depicted in these videos remains unchanged.

3.4.6 StyleGAN image generator pre-trained on FFHQ dataset

In the previous section, we show the results of the framework where the image encoder and StyleGAN image generator were pre-trained on video frames of talking-head datasets. In this section, we experiment with an image encoder and StyleGAN image generator that are pre-trained on FFHQ which contains faces with static expressions (i.e., not talking) while the latent generator and video discriminator are trained with the talking-head video datasets (TCD-TIMIT and GRID). The image encoder is the pSp encoder we discuss in Section 3.16 and the image generator is the same one we discuss in Section 3.3.1.1. Both models were pre-trained on the FFHQ dataset. In this case, the identities of the training talking-head dataset have not been seen by the image encoder and the image generator. We have seen in Section 3.3.2.1 that projecting video frames using the pSp encoder pre-trained on the FFHQ dataset results in reconstructed frames that exhibit some appearance distortion (Figure 3.17); however, the facial expressions are preserved. Therefore, training the same framework (Figure 3.23) with the video discriminator in Eqn. 3.7 would result in the latent generator G_l being optimized to not only generate plausible motions but also improve the visual quality. This is because the



(a) Video samples generated using the unconditional video generation model trained on GRID dataset

(b) Video samples generated using the unconditional video generation model trained on TCD-TIMIT dataset

Figure 3.24: Video samples generated using the unconditional video generation model. The model employs an image encoder and a StyleGAN image generator pre-trained on video frames from talking-head datasets: (a) GRID and (b) TCD-TIMIT. We can observe that the model is capable of generating video frames that exhibit talking expressions.

Figure 3.25: Video samples were generated using the unconditional video generation model presented in Figure 3.23, with video discriminator loss 3.7. The model employs an image encoder and a StyleGAN image generator pre-trained on the FFHQ dataset of static head images. It is clear that the latent sequence generator struggles to produce convincing talking-like expressions, instead prioritising the enhancement of visual quality.

video discriminator classifies real video samples and fake samples. The real samples from the training dataset have no distortion while the fake video samples exhibit visual distortion. Figure 3.25 shows generated samples using the framework trained with the video discriminator loss in Eqn. 3.7. It can be seen that the latent generator G_l fails to generate plausible talking-like expressions but instead focuses on improving visual quality.

To overcome the aforementioned issue, we propose to use the projected real video as the real video samples to train the video discriminator. We first project the real video frames using the pSp image encoder into the latent space of the pre-trained StyleGAN image generator to reconstruct the real video frames. Then, the latent generator is optimized with the video discriminator on both the real projected videos as well as the fake generated video frames. The updated framework overview is shown in Figure 3.26. Using the projected videos as the real video, the objective loss is therefore as follows:

$$\min_{G_l}\max_{D_v}\mathcal{L}_{\mathcal{VD}}(D_v, G_l) = \mathbb{E}_{X\sim p_{\text{data}}}[\log D_v(G(E_I(X)))] + \mathbb{E}_{I\sim p_{\text{data}}}[\log(1 - D_v(G(G_l(n, E_I(I)))))]. \tag{3.10}$$

Figure 3.27 shows generated video samples using the framework trained with the image encoder

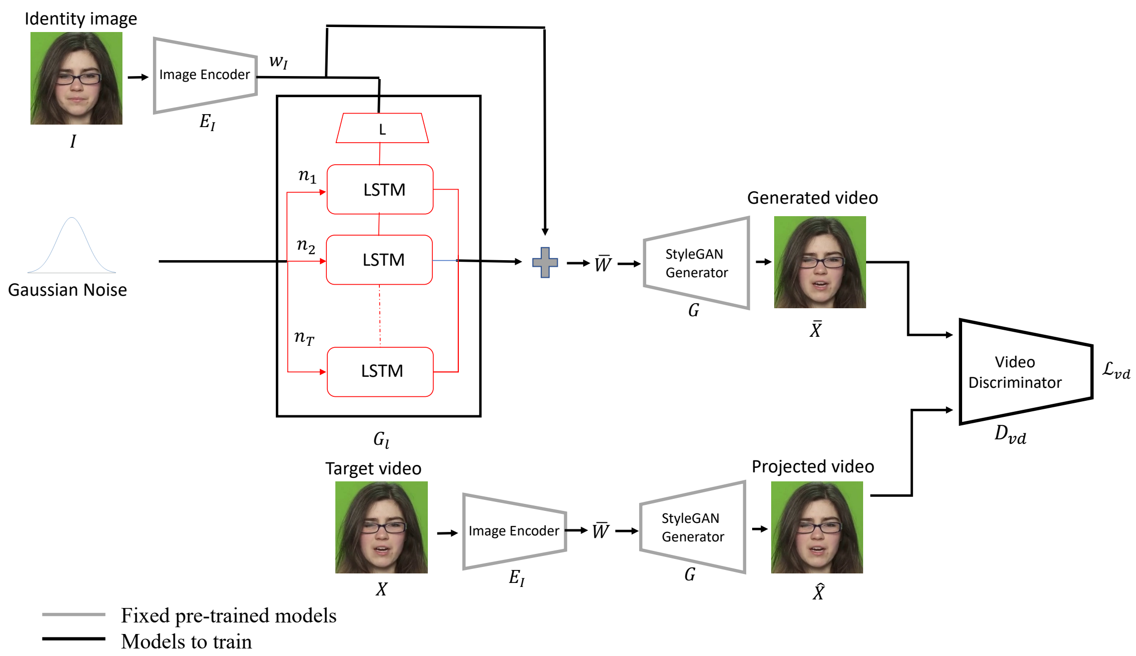
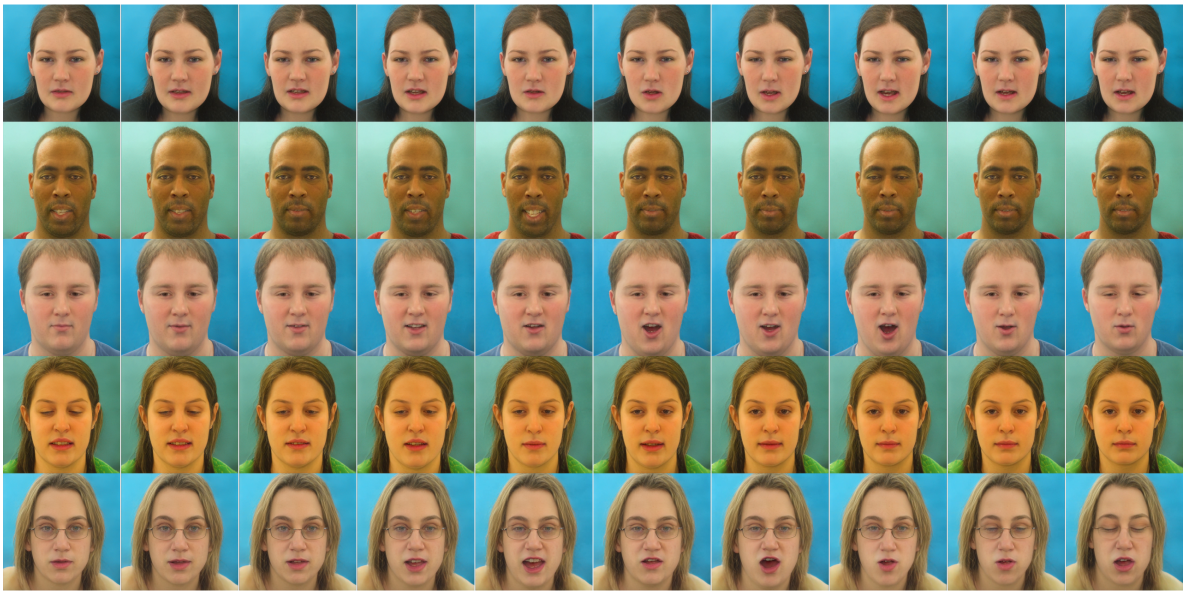


Figure 3.26: A modification of the framework presented in figure 3.23. The video discriminator takes as input the projected real video as well as the generated one.

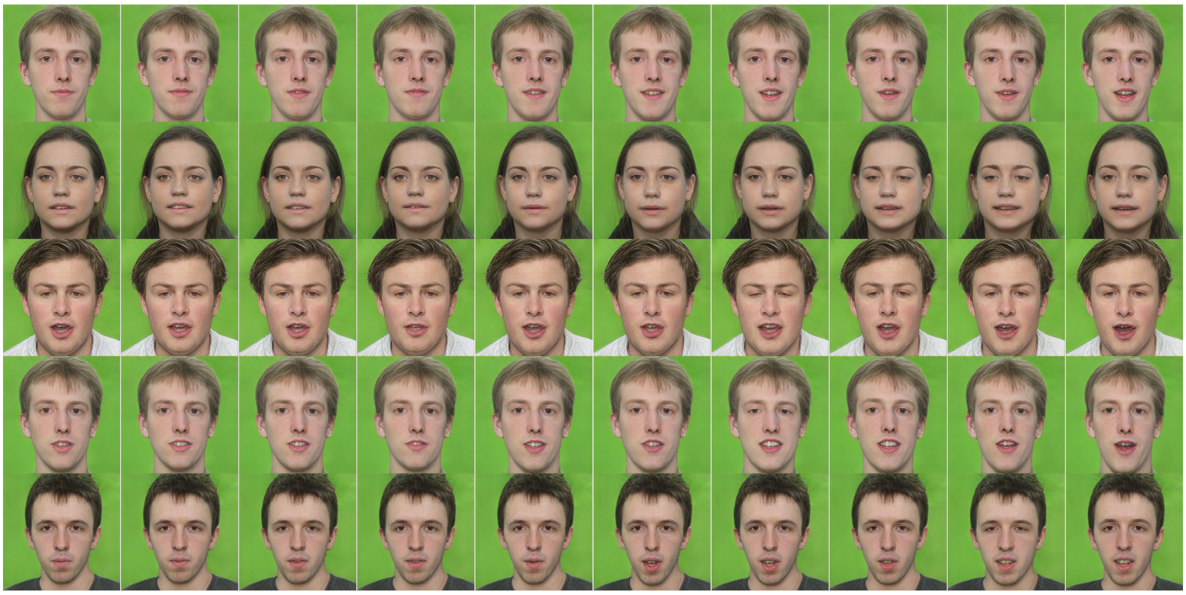
E_I and StyleGAN image generator G pre-trained on the FFHQ dataset. We can see that the model is able to generate diverse motions of mouth movements and other facial expressions such as eye blinks. In training, the model takes an identity image to set the identity of a video and noise samples which drive the motion of the video. The model learns to disentangle the motion from the appearance during training without explicitly learning it. Figure 3.28 illustrates how the appearance and the motion are disentangled. In Figure 3.28b, video samples are generated with the same identity image i.e. same appearance while varying the input noise samples i.e. various motions. In Figure 3.28a, we show video samples with the same motion while varying the identity images. This property is very useful for transferring motions from one identity to another.

Our model can generate videos from any identity since the image encoder and the StyleGAN image generator are pre-trained on the FFHQ dataset over which a general appearance model emerges. We can take any identity image from the wild to the initial appearance of the video. In Figure 3.29, video samples are generated from initial identity images collected from the wild. The model can also generate videos from input images of portrait paintings (seen in the 3rd and 5th rows).

We compare the results of our framework with the results of the MoCoGAN-HD model [Tian

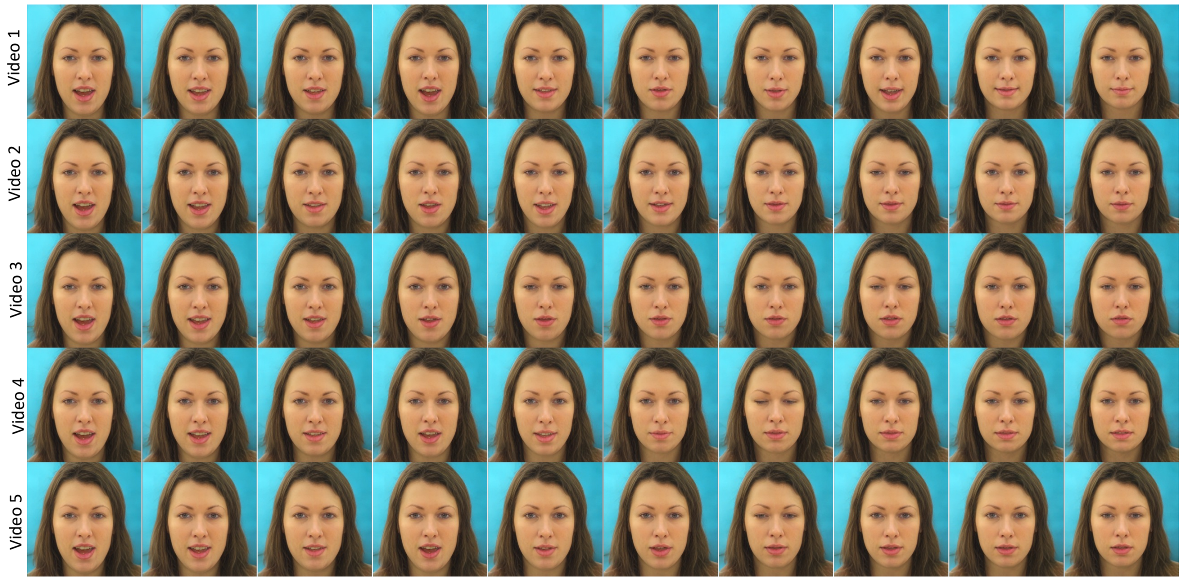


(a) Video samples generated using the unconditional video generation model trained on GRID dataset



(b) Video samples generated using the unconditional video generation model trained onTCD-TIMIT dataset

Figure 3.27: Video samples generated using the unconditional video generation model trained with the image encoder E_I and the StyleGAN image generator G, pre-trained on the FFHQ dataset of static head images. The latent generator G_l and the video discriminator D_v are trained with the novel loss presented in Eq. 3.10 on video frames of talking heads.



(a) Video samples were generated using the same input identity image but with varying input noise sequences.

(b) Video samples were generated using the same input noise sequence but with varying input identity images.

Figure 3.28: Video samples were generated using the unconditional video generation model to demonstrate that the model learns to disentangle motion and appearance in the generated videos. The model is trained on video frames of talking heads with the image encoder and the StyleGAN image generator, which were pre-trained on images of static heads.

Figure 3.29: Video samples were generated using the unconditional video generation model with identity images taken from the wild. The model is capable of generating plausible videos featuring a variety of expressions.

et al. 2021]. Similar to our framework, the MoCoGAN-HD utilises the latent space of the StyleGAN image generator pre-trained on the FFHQ dataset of static heads to generate videos from a noise sequence. It samples a sequence of noise vectors and a latent code from the latent space and trains a latent sequence generator to output a sequence of latent codes which is then used to generate a video using the StyleGAN generator. In addition, both models train with the same video discriminator loss and information loss (see Section 3.4.3). However, our framework and the MoCoGAN model differ in the following:

- The MoCoGAN-HD model generates a sequence of latent vectors in the W intermediate latent space whereas our framework works in the extended W^+ intermediate latent space. The W^+ is a bigger space than the W space and it is more disentangled, making it very suitable for manipulating a latent code of a given image. In addition, we found that projecting videos in the W^+ latent space produces a better result than the W latent space, as can be seen in Figure 3.5.

- One major difference is that our framework uses an image encoder to encode a given identity image to set the initial appearance for the generated video whereas MoCoGAN-HD sets the initial appearance by randomly sampling a latent code from the latent space of the pre-trained StyleGAN. The depicted identity in the generated video by MoCoGAN-HD is always different from the identities of the training talking-head video datasets as the image generator is pre-tranied on the FFHQ dataset. Therefore, the MoCoGAN-HD model requires an additional contrastive loss to help train the latent sequence generator as the video discriminator takes as input two videos depicting different identities. In our framework, the identity image is from the training talking-head video dataset and therefore the input videos of the video discriminator depict the same identity, making the video discriminator focus on learning to generate plausible talking motion.

- The output of the latent sequence generator of our framework recurrently displaces only the identity latent code whereas the MoCoGAN-HD model displaces the generated latent code of the previous time step. We found our approach keeps the identity of the generated video consistent throughout the video. The MoCoGAN-HD model uses an addition image-based discriminator loss that takes as input the frame of the sampled latent code as the real sample and each generated frame as the fake sample to help maintain the identity throughout the generated video.

Figure 3.30 presents a comparison between the results of our framework and those of the MoCoGAN-HD model. The MoCoGAN-HD results are adapted from the model's project webpage[3]. We use the first frame of each video generated by the MoCoGAN-HD model as the identity image and employ our framework to generate the corresponding videos for comparison. The figure illustrates that our framework produces more realistic talking-like motions than MoCoGAN-HD, which is particularly highlighted in yellow. Additionally, our framework is capable of generating diverse non-lip facial movements.

3.5 Conclusion

In this chapter, we studied the latent space of a pre-trained StyleGAN image generator for video projection and generation. For video projection, we specifically examined two GAN inversion methods: optimisation-based (Section 3.3.1) and encoder-based methods (Section 3.3.2). We found that the optimisation-based method produces reconstructed video frames that are very similar to real video frames. However, the method is considerably slower compared to the encoder-based method. We also demonstrated that the encoder-based method generates plausible video frames while being very fast. This makes the encoder-based method to be preferable for online downstream tasks such as latent space manipulation.

[3] `https://bluer555.github.io/MoCoGAN-HD/`

(a) Video generated using our unconditional video generation framework

(b) Video generated using the MoCoGAN-HD model

Figure 3.30: The figure illustrates videos generated by our unconditional video generation framework in comparison with the MoCoGAN-HD model. It is evident from the figure that our framework can generate more diverse talking-like motions than the MoCoGAN-HD model, as highlighted in yellow.

In the second part of this chapter, we used the learned encoder-based method to build a video generative model. We proposed a video generative model that uses a pre-trained StyleGAN image generator to generate videos from temporal noise samples. The idea is to map an identity image to the latent space of the pre-trained image generator using the learned encoder-based method (Section 3.3.2) and deviate its latent code to generate a trajectory of latent codes conditioned on the noise samples. We showed that by using the encoder-based projection method and the pre-trained image generator, the trained video generative model can generate realistic talking-head videos with different motions. We also demonstrated that the model learns to disentangle motion from appearance without explicitly training it to do so.

In the video projection and generation sections, we experimented with two versions of StyleGAN; one trained on video frames depicting talking heads and the other on faces with static expressions (i.e., not talking). The resulting videos from projecting videos into the latent space of the StyleGAN pre-trained on talking-head video frames using the image encoder look very similar to the input videos. However, the image encoder cannot project any video that has not been seen (Figure 3.22).

We demonstrate that using the pSp encoder [Richardson et al. 2021] we can project any video into the latent space of the StyleGAN pre-trained on the FFHQ dataset. This is because the encoder was trained to project images from the FFHQ dataset, which contains a large number of images, allowing the encoder to generalise to any image from the wild.

One might suggest training the StyleGAN model on a combination of datasets consisting of static images and talking-head video frames. Such a model could benefit from the varied expressions found in the talking-head video datasets, while also allowing our approach to generalize through training on the static images. However, we have explored this idea and found that the StyleGAN model struggles to generate plausible images. We believe this may be due to the fact that the two types of datasets originate from different domains and, consequently, exhibit distinct underlying data distributions.

In the next chapter, we build upon the findings of this chapter to propose an approach that generates talking-head videos from speech audio. Specifically, the approach is similar to the unconditional video generation approach presented in the second part of the chapter (Section 3.4). We use the learned image encoder to project an identity image into the latent space of the pre-trained StyleGAN image generator. We then train a recurrent model that takes as input the speech audio and displaces the obtained identity latent code to generate a sequence of latent codes. We know from the results of the unconditional video generation (e.g., Figures 3.27 and 3.29) that the latent space of StyleGAN is suitable for video generation.

Chapter 4

Audio-driven Video Generation

4.1 Introduction

In the previous chapter, we learned an image encoder that projects a talking-head video into the latent space of a pre-trained StyleGAN image generator. We then used this encoder to train a generative model that generates plausible trajectories of a talking head in the latent space. We showed that the image encoder experimentally keeps the generated trajectories within the subspace of a single identity. In this chapter, we propose an approach that utilises the learned image encoder model to find a latent code of an identity frame and maps speech audio to a trajectory in the latent space of a pre-trained StyleGAN image generator, rooted at the latent code of an identity frame. These generated latent trajectories are then used to generate corresponding talking-head videos using the image generator.

We train two generative models that utilise a pre-trained StyleGAN image generator. In the first model, the StyleGAN image generator is pre-trained on frames of talking-head videos while the second model uses a StyleGAN image generator pre-trained on images of faces with static expressions (i.e., not talking). For training both generative models, we use two audio-visual talking head datasets: TCD-TIMIT and GRID. Our results demonstrate that the approach enables us to generate photo-realistic talking-head videos from speech. We also show that we can drive the generation using any speech audio from the wild.

4.2 The image generator and encoder pre-trained on video frames of talking heads (first model)

In this model, we experiment with a StyleGAN image generator [Karras et al. 2019] that is pre-trained on video frames of a talking-head dataset. Specifically, we use the pre-trained image generator that was introduced in Section 3.3.1.2. This same pre-trained StyleGAN image generator was also used for video projection (Section 3.3.2.2) and unconditional video generation (Section 3.4.5). Additionally, we utilise the pre-trained image encoder that was introduced for video projection in Section 3.3.2.2. In the following, we introduce the framework of our first model and the training strategy, followed by implementation details.

4.2.1 Framework

Our framework consists of six components: image encoder E_I, audio encoder E_A, latent encoder E_L, decoder D, noise generator G_{noise} and image generator G. Given an identity image I and speech audio a, the goal is to synthesise a video of the identity lip-synced with the audio. We first partition the audio clip into a sequence of T audio segments $\{a_1, a_2, .., a_T\}$ (320 ms for each segment in our experiments). From this audio segment sequence, we target a video clip consisting of a sequence of T video frames $\{x_1, x_2, .., x_T\}$. There is therefore a one-to-one correspondence between input audio segments and output video frames.

The inference pipeline is as follows. We take the identity image I and find its latent code w_I in the latent space W^+ of the generator G using the image encoder E_I. Next, we encode an audio segment a_t using the audio encoder E_A and extract an audio embedding vector $e_t = E_A(a_t)$. We encode the identity latent code w_I to reduce its high dimensionality and feed it with the audio vector e_t jointly along with a temporal noise sample n_t to the decoder D to predict a latent displacement $d_t = D(w_I, e_t, n_t)$. We then calculate the displaced latent vector $\bar{w}_t = d_t + w_I$. Lastly, the image generator G takes the displaced latent vector $\bar{w}_t$ and generates the corresponding video frame $\bar{x}_t$. An overview of the model can be seen in Figure 4.1. In the following section, we describe each component in detail.

4.2.1.1 Image Generator

The image generator takes as input a latent code $\bar{w}$ and generates corresponding video frame $\bar{x}$. In this framework, we use the StyleGAN image generator we discuss in section 3.3.1.2.

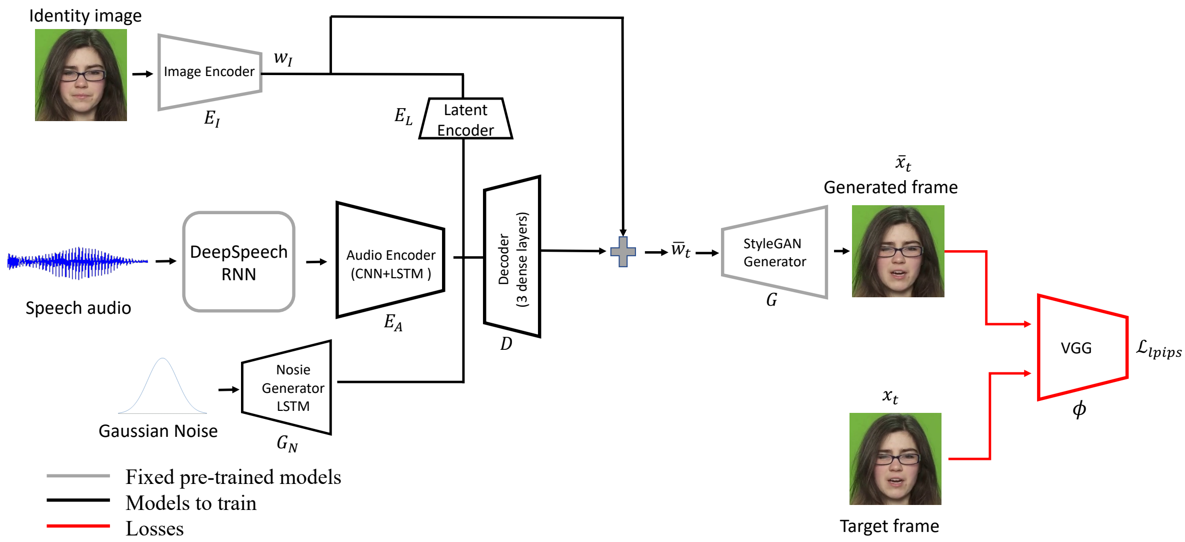


Figure 4.1: Overview of the model: Given an identity image and a speech audio, the aim is to synthesise a video of the identity lip-synced with the audio. We first find the corresponding latent code of the identity image w_I using the image encoder E_I and then reduce its dimensionality using the latent encoder E_L. We then encode the audio using the audio encoder E_A. Next, we generate temporal noise n_t using the noise generator G_N. The decoder D then takes the reduced identity latent vector, encoded audio and the noise to predict a displacement to the identity latent vector in the latent space of a pre-trained image generator G.

4.2.1.2 Image Encoder

The goal of the image encoder E_I is to map a given identity image I to the latent space W^+ of a pre-trained generator G. We use the image encoder we introduced in the previous chapter, see Section 3.3.2.2.

4.2.1.3 Audio Encoder.

The audio encoder E_A consists of two networks. The first network maps from an audio feature representing each audio segment to an embedding vector. We represent audio segments using the DeepSpeech model [Hannun et al. 2014], which we explain in detail in Section 4.2.3. We use the DeepSpeech representation to increase the robustness of our model to various audio inputs, as the DeepSpeech model is trained on a very large audio dataset encompassing different recording environments, which may include recording artefacts and background noise. Similar to Cudeiro et al. 2019; Thies et al. 2020b , this network consists of 4 1d-convolution layers each followed by a leaky ReLU (0.02). The second network is a recurrent model with a 3-layer LSTM that maps from the embedding audio vector to another embedding vector of the same size. The purpose of using LSTM units is to capture the temporal dependencies within audio sequences, which is essential for generating coherent talking-head videos. The architecture details of the audio encoder are described in Table 4.1.

Layer type	Kernel	Stride	Output	Activation
1d convolution	3	2	32x8	Leaky ReLU
1d convolution	3	2	32x4	Leaky ReLU
1d convolution	3	2	64x2	Leaky ReLU
1d convolution	3	2	64x1	Leaky ReLU
3-layer LSTM	-	-	64x1	-

Table 4.1: The table describes the architecture of the audio encoder in the first model. The slope of Leaky ReLU is set to 0.02 in all layers.

4.2.1.4 Latent Encoder.

Our model predicts the displacement to the encoded identity in the latent space of the pre-trained StyleGAN image generator G. Therefore, the model needs to know the latent space point from which the displacement is applied. We use a latent encoder E_L consisting of a single fully connected layer with size 128 to encode the identity latent code. The layer weights are initialised with the first 128 pre-computed Principal Component Analysis (PCA) components. The PCA is performed on the latent codes of the training video frames from the talking-head dataset within the latent space of the image generator G. These latent codes are obtained using the image encoder E_I.

Layer type	**Output**	**Activation**
	Latent encoder	
linear	128	-
	Noise generator	
1-layer LSTM	10	-
	Decoder	
linear	256	ReLU
linear	256	ReLU
linear	256	ReLU
linear	6144	ReLU

Table 4.2: The table describes the architecture of the latent encoder, noise generator, and decoder in the first model

4.2.1.5 Noise Generator.

Although the audio has all the information needed to drive the lip movements in the latent space, other spontaneous face expressions e.g. blinks are in less degree dependent on the audio. Moreover, audio-driven facial animation is a one-to-many task which means there are multiple outputs for a single audio speech [Richard et al. 2021b]. Inspired by Vougioukas et al. 2018a, we add a noise generator G_N that generates a temporal noise to accommodate spontaneous face expressions that in a high degree are not correlated with the speech. The network consists a single LSTM unit with size 10. It takes a 10-dimensional vector sampled from a Gaussian distribution and produces a sequence of temporal noise n_t.

4.2.1.6 Decoder

The output of the audio encoder, latent encoder, and noise generator are concatenated, resulting in a vector of 202 dimensions. The decoder network D has 3 fully connected layers each with a size of 256 except for the output layer which matches the latent space W^+ dimension. Each of the first two layers is followed by a leaky ReLU (0.02). We initialise the first layers randomly while initialising the output layer with the inverse of the first 256 PCA components of the latent space W^+, which is pre-computed in Section 4.2.1.4.

4.2.2 Training strategy

Training the model is done in three stages. In stage 1, we train the StyleGAN image generator on the video frames of talking heads. In stage 2, we train the image encoder to project video frames into the latent space of the pre-trained StyleGAN image generator from stage 1. These two training stages follow the same methodology as discussed in the previous chapter (stage 1 in Section 3.3.1.2 and stage 2 in Section 3.3.2.2). However, we provide an explanation of these training stages here for this section to be self-contained. In stage 3, we train the other components of our framework i.e. audio encoder E_A, latent encoder E_L, decoder D and noise generator G_{noise}. The rationale behind dividing the training into three stages is to reduce the overall complexity of the model training process. Stages 1 and 2 focus on learning the appearance of talking-head videos, whereas Stage 3 is to learn the underlying dynamics of the videos.

4.2.2.1 Stage 1: training the image generator

The image generator is the synthesis network of the StyleGAN model. The StyleGAN image generator is trained on the video frames of a talking head dataset. To ease the training, we train further all the layers of the pre-trained StyleGAN model which was trained on the FFHQ dataset [Karras et al. 2019] on our training samples. We use the same configuration (config-F) used in [Karras et al. 2019] for training. We train the StyleGAN model on two talking-head datasets: GRID [Cooke et al. 2006] and TCD-TIMIT [Harte and Gillen 2015]. We measure the performance of the training using the FID score as the most common metric for image generation tasks. The trained synthesis network of the generator is used in the following stages.

4.2.2.2 Stage 2: training the image encoder

The image encoder E_I is trained in an auto-encoder manner where the decoder is the pre-trained generator G from stage 1. To ease the training, the encoder only predicts the displacement Δw to the mean latent code $\bar{w}$. The mean vector is calculated by sampling from W^+ of the trained StyleGAN model from stage 1. We train only the encoder on the same datasets (i.e. GRID and TCD-TIMIT) in stage 1 while keeping the decoder fixed. Figure 3.19 shows the training process of the image encoder. For the loss function, we use an L_2 loss to measure the difference between an input image x and reconstructed image $\bar{x}$. Additionally, we use a perceptual loss [R. Zhang et al. 2018] between the two images. The loss for the second stage is as follows:

$$\mathcal{L}_{\text{stage}_2} = ||x - G(E_I(x))||_2 + ||\phi(x) - \phi(G(E_I(x)))||_2 \tag{4.1}$$

where ϕ is a VGG network [Simonyan and Zisserman 2015].

4.2.2.3 Stage 3: training the model

In this stage, we use the pre-trained StyleGAN image generator from stage 1 and the pre-trained image encoder in our model as fixed components while training other components. We train these components on the same two talking-head datasets we used in Stage 1 and 2, i.e., GRID and TCD-TIMIT. For the loss function, previous works use a combination of multiple losses in training a talking-head generation model. Some works use adversarial losses [Vougioukas et al. 2018a; Vougioukas et al. 2019; L. Song et al. 2020; Y. Zhou et al. 2020b] such as a sequence and

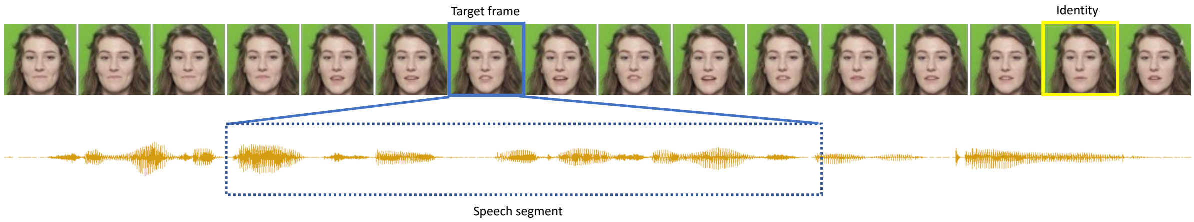


Figure 4.2: Sampling strategy of audio segments and identity images in the first model. The identity images are randomly sampled from out of the audio sample window. The target frame is selected to be the fourth frame of the audio window.

frame discriminators to improve the realism of the generated videos. Some prior work uses the pixel-wise L_1 loss as an additional reconstruction loss to capture the correct mouth movements [Vougioukas et al. 2018a; Vougioukas et al. 2019]. L_1 loss is known to produce blurry images when it is used as a single loss [Chung et al. 2017b]. To make the training of our model stable, we use only a perceptual loss [Johnson et al. 2016; R. Zhang et al. 2018] as our single loss function to improve the visual quality of the generated videos. The perceptual loss computes how similar a generated frame $\bar{x}_t$ is to the target frame x_t in a way similar to human judgement. The loss is the sum of the perceptual loss on each frame as follows:

$$\mathcal{L}_{\text{stage}_3} = \sum_{t=1}^{T} ||\phi(\bar{x}_t) - \phi(x_t)||_2 \tag{4.2}$$

4.2.3 Dataset

We use two datasets for synthesizing talking-head videos: GRID [Cooke et al. 2006] and TCD-TIMIT [Harte and Gillen 2015]. These are the same datasets we use to train the video generative model we discuss in Section 3.4. For video pre-processing, we follow the process discussed in Section 3.3.1.1. Each audio segment has a length of 320 ms, which corresponds to a window of eight video frames. The audio segments overlap with a slide of a single frame, meaning that we move the window by a single frame. We choose only the fourth frame as the ground truth frame, following the same choice as in [Thies et al. 2020b]. The identity image is a randomly chosen frame from this window. This enables us to use any arbitrary identity image as the starting point in the latent space of GAN. Figure 4.2 shows the sampling strategy of a single data sample.

Audio feature extraction. We represent the audio input using the character logits of the pre-trained speech-to-text DeepSpeech RNN model [Hannun et al. 2014]. The model was trained and generalized across a large number of audio speakers. Therefore, our model is capable to synthesise videos from novel speech audio. Similar to Cudeiro et al. and Thies et al. [Cudeiro et al. 2019; Thies et al. 2020b], we use a window of 16 intervals each of length 20 ms to extract the audio features from the DeepSpeech model. The output of the model is unnormalised log probabilities of characters with a length of 29. Therefore, the audio feature is an array of shape 16 x 29. The audio windows are sampled with a stride of 4 ms.

4.2.4 Implementation

We implemented our experiments using the deep learning framework PyTorch [Paszke et al. 2019] and ran them on a GPU with a 32GB memory size. We use an unofficial implementation of StyleGAN [1] in PyTorch and its accompanying pre-trained models. **Stage 1.** We train the pre-trained StyleGAN further on our training video frames for 3 epochs. **Stage 2.** We train the image encoder with a learning rate of 0.004 for four epochs. **Stage 3.** We train the first model for 20 epochs and use Adam optimiser [Kingma and Ba 2014] with a 0.0002 learning rate. We decay the learning rate when the metric does not improve by a factor of 10. The generator G and the image encoder E_I are both kept fixed while training other components.

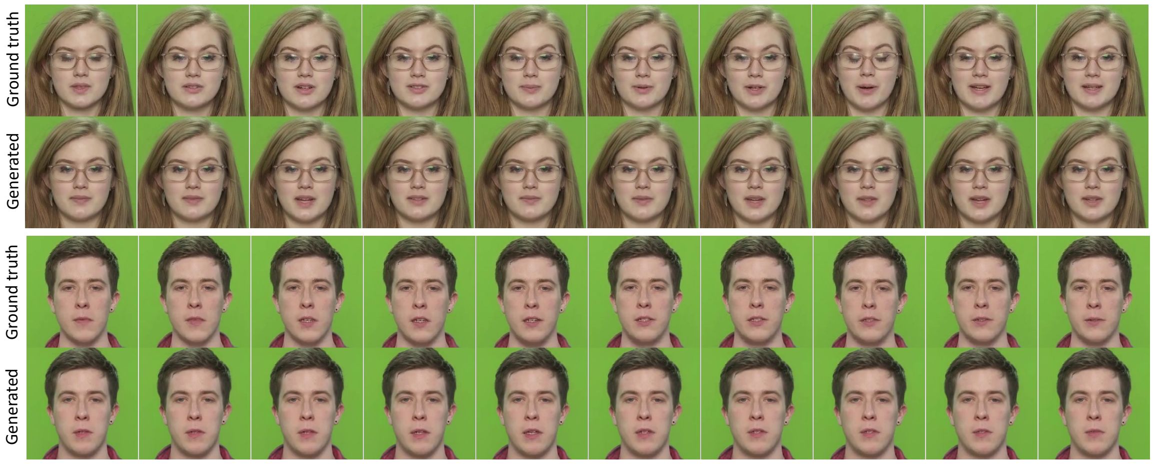
4.2.5 Results

We train the first model on two datasets: GRID and TCD-TIMIT. We demonstrate the performance of the model by showing generated videos driven by unseen speech audio from the training identities. Figure 4.3 shows generated videos using the first model.

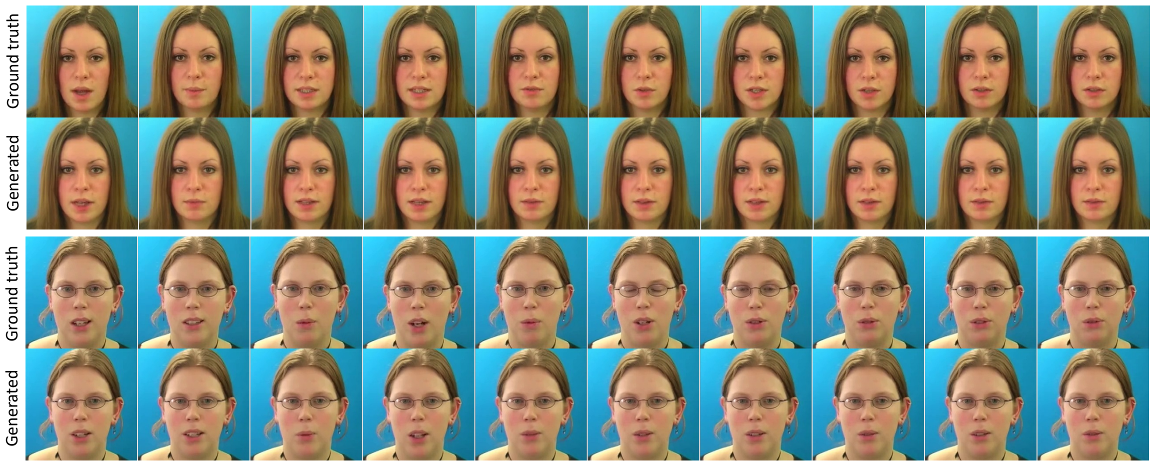
Cross generation. We test our first model using the audio speech of one identity to drive another. We drive the video generation of a seen identity using an audio sample of an unseen identity from the same dataset. Figure 4.4 shows the quality of the generated videos on an unseen audio sample. We can see that the model generates videos with accurate mouth movements similar to the ground truth video. The figure demonstrates the robustness of the model to handle speech audio of unseen speakers. This can be attributed to the fact that the DeepSpeech model [Hannun et al. 2014], the audio presentation, was trained on a very large audio dataset, enabling our first model to generalise to unheard speech audio.

Unseen Identities. We test the performance of the model by using identities that were not

[1] https://github.com/rosinality/stylegan2-pytorch.

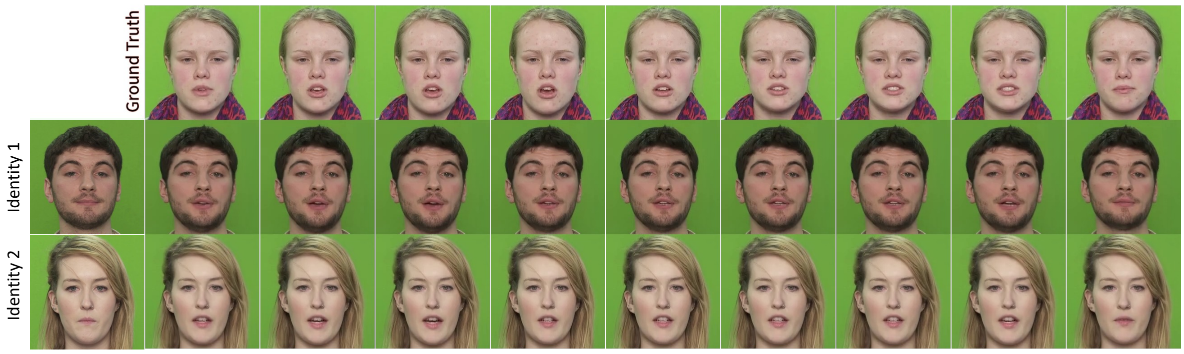


(a) Video samples generated from TCD-TIMIT dataset



(b) Video samples generated from GRID dataset

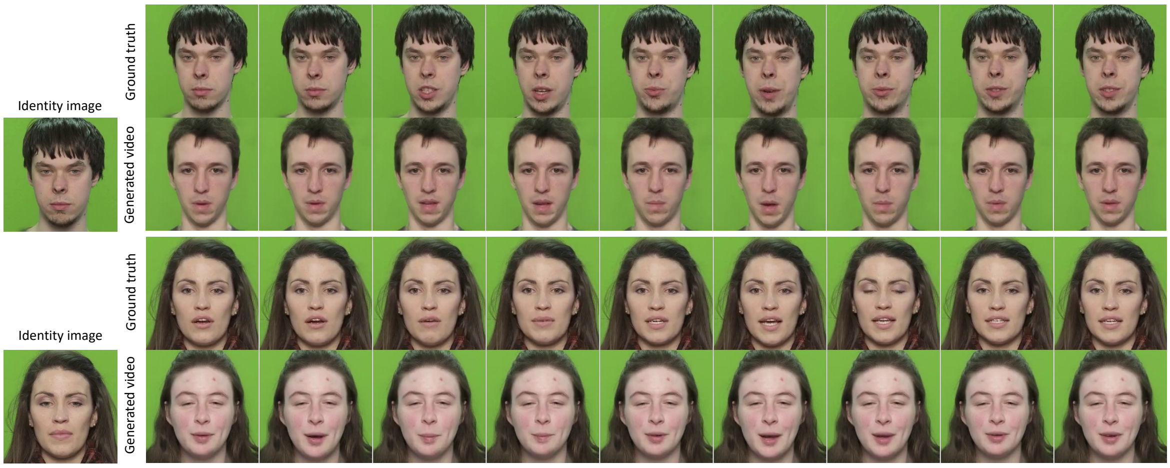
Figure 4.3: Generated videos of our first model. We generated the videos using unseen speech audio from one of the training identities: (a) TCD-TIMIT and (b) GRID. For each sub-figure, the first row represents the ground truth frames of the input speech audio while the second row shows the generated frames



(a) Video samples generated from TCD-TIMIT dataset

(b) Video samples generated from GRID dataset

Figure 4.4: The outputs of our first model. We generated the videos using unseen audio inputs of unseen identities selected from two datasets: (a) TCD-TIMIT and (b) GRID. For each sub-figure, the first row represents the ground truth frames of the input speech audio. The other two rows show the generated frames. From this figure, our model synthesises videos well-synced with the input audio. It can be seen in the last column from each sub-figure that our model generates a non-open mouth when there is no spoken word



(a) Video samples generated from TCD-TIMIT dataset

(b) Video samples generated from GRID dataset

Figure 4.5: videos generated of unseen identities. We can see the first model fails to preserve the given identity. However, the generated videos have lip movements synced with the target video.

present in the training set. Figure 4.5 shows generated videos of these unseen identities. From the figure, we can see the first model is unable to accurately preserve the given identity image in the generated video. Instead, the generated videos present one of the training identities that the pre-trained encoder was previously trained on. However, the generated videos have plausible talking-head videos with synchronized mouth movements corresponding to the given audio. This suggests that the image encoder has learned to map the provided identity image to the latent code of the most similar training identity. Furthermore, the model has successfully maintained consistency in the identity across all frames of the generated video.

4.3 The image generator and encoder pre-trained on images of static heads (second model)

In the first model, we experiment with the StyleGAN image generator and the image encoder, both of which are pre-trained on video frames of talking heads. In contrast, the second model involves experimenting with them being pre-trained on images of stationary heads. The pre-trained image generator and image encoder we use in this second model were introduced in the previous chapter, specifically in Sections 3.3.1.1 and 3.3.2.1, respectively. The image encoder is trained to project images into the latent space using the FFHQ dataset which contains a large number of images of human faces. This allows our second model to project an identity image from the wild to generate a talking-head video of the projected identity. In addition, the pre-trained StyleGAN image generator of our second model can generate high-resolution images of size 1024x1024. This is due to the image generator being trained on the FFHQ dataset which consists of high-resolution images. Incorporating this image generator in our framework enables the second model to generate high-resolution videos.

In the second model, we aim to generate high-resolution videos from any given input identity image. Thus, the framework of this model differs from the first in three aspects. Firstly, the input audio to the audio encoder is represented using the MFCC features, rather than the DeepSpeech features. This is because DeepSpeech features are character logits, representing only the linguistic content and not the prosody of the speech. A talking-head model capable of interpreting and reproducing prosody nuances can more accurately generate facial expressions that match the speech's tone. Consequently, the architecture of the audio encoder in our second model is different from that in the first model.

Secondly, the loss functions for training the model are different from those in the first model, as we will justify in the following sections. Thirdly, in the second model, we do not use a noise generator and latent encoder. This is because using the noise generator would require training the model with an additional adversarial loss, such as a video discriminator, which would increase the computational cost of training. Lastly, the latent encoder, introduced in the first model to reduce the high dimensionality of the identity latent code, is replaced in the second model by PCA. This change serves two purposes: to reduce the number of learnable parameters and to help the model generalize to any identity given.

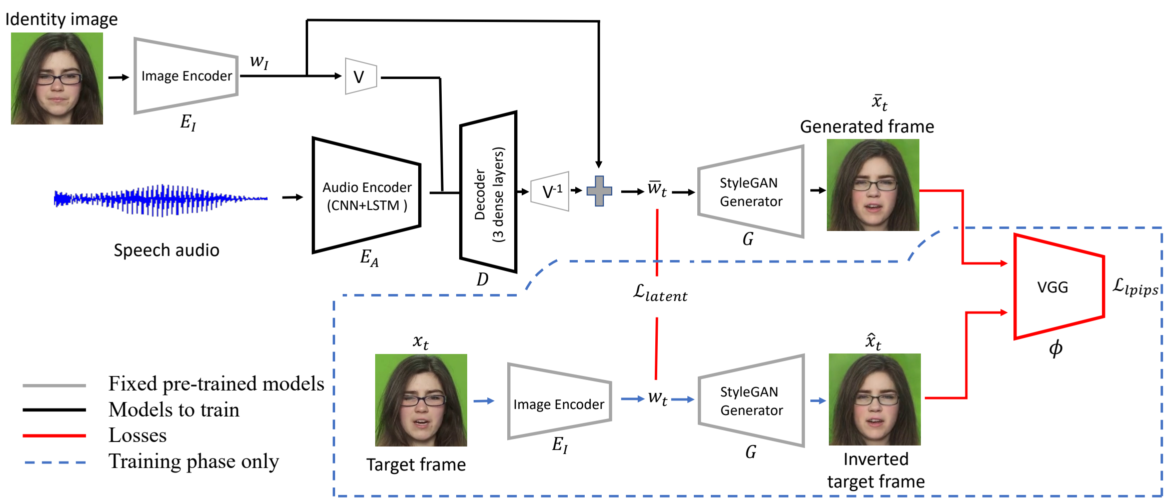


Figure 4.6: Overview of the model: Given an identity image and speech audio, the aim is to synthesise a video of the identity lip-synced with the audio. We first find the corresponding latent code of the identity image using the image encoder E_I. We then encode the audio using the audio encoder E_A. Next, we embed the identity latent code using the PCA basis V. The decoder D then takes both the embedded identity latent vector and encoded audio to predict a displacement to the identity latent vector in the latent space of a pre-trained image generator G.

These changes are made independently of each other and help to make the second model generalizable and generate high-resolution videos. The following sections will introduce the framework and training strategies for this model, along with implementation details and results.

4.3.1 Framework

The framework consists of four components: image encoder E_I, audio encoder E_A, latent decoder D and image generator G. Given an identity image I and speech audio a, the goal is to synthesise a video of the identity lip-synced with the audio. We first partition the audio clip into a sequence of T fixed-duration audio segments $\{a_1, a_2, .., a_T\}$. From this audio segment sequence, we target a video clip consisting of a sequence of T video frames $\{x_1, x_2, .., x_T\}$. There is therefore a one-to-one correspondence between input audio segments and output video frames.

The inference pipeline is as follows. We take the identity image I and find its latent code w_I in the latent space W^+ of the generator G using the image encoder E_I. Next, we encode an audio segment a_t using the audio encoder E_A and extract an audio embedding vector $e_t = E_A(a_t)$. We then feed in the identity latent code w_I and the audio vector e_t jointly to the latent decoder D to predict a latent displacement $d_t = D(w_I, e_t)$. We then calculate the displaced latent vector

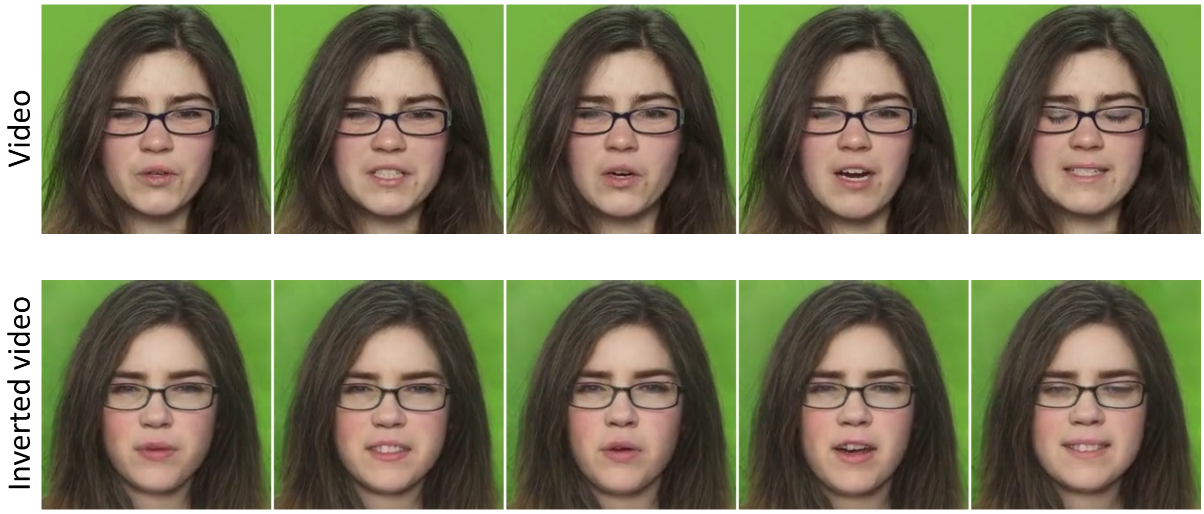


Figure 4.7: This figure illustrates how the image encoder E_I can encode a video (top row) into the latent space of StyleGAN W^+ while preserving its mouth movements and other facial expressions. Bottom row shows the inverted video.

$\bar{w}_t = d_t + w_I$. Lastly, the image generator G takes the displaced latent vector $\bar{w}_t$ and generates the corresponding video frame $\bar{x}_t$. An overview of the model can be seen in Figure 4.6. In the following section, we describe each component in detail.

4.3.1.1 Image generator

The image generator takes as input a latent code $\bar{w}$ and generates corresponding video frame $\bar{x}$. In this second model framework, we use the StyleGAN image generator we discuss in Section 3.3.1.1. This is in contrast to the first model where the image generator is pre-trained on video frames of talking heads.

4.3.1.2 Image encoder

We use pSp [Richardson et al. 2021], an off-the-shelf pre-trained image encoder that inverts real images to the W^+ space of StyleGAN. The encoder was trained by projecting the FFHQ [Karras et al. 2020] dataset to a fixed StyleGAN generator. This image encoder E_I is the same as the image encoder we discuss in Section 3.3.2.1. Critically, the encoder produces latent vectors that preserve the mouth expression on synthesis with StyleGAN as it can be seen in Figure 4.7. The figure shows video frames that have been projected into the latent space W^+ using the image encoder E_I and then re-generated using G.

4.3.1.3 Audio encoder

We represent each audio segment a_t using Mel-frequency cepstral coefficients (MFCC). The audio encoder E_A is a recurrent model that takes a sequence of audio segments $\{a_1, a_2, .., a_T\}$ and produces a sequence of encoded audio segments $\{e_1, e_2, .., e_T\}$. The encoder network E_A consists of multiple convolutional layers followed by three LSTM layers.

Layer type	Kernel	Stride	Output	Activation
2d convolution	3x3	1	64x12x28	ReLU
2d convolution	3x3	1	128x12¡x28	ReLU
2d max pooling	3x3	1x2	128x10x13	-
2d convolution	3x3	1	256x10x13	ReLU
2d convolution	3x3	1	256x10x13	ReLU
2d convolution	3x3	1	512x10x13	ReLU
2d max pooling	3x3	2x2	512x4x6	-
linear	-	-	2048	ReLU
linear	-	-	256	ReLU
3-layer LSTM	-	-	256	-

Table 4.3: The table describes the architecture of the audio encoder in the second model

4.3.1.4 Latent decoder

The latent decoder D takes as input the identity latent vector w_I and an encoded audio segment e_t to predict a displacement d_t to the identity vector in the latent space W^+. To reduce the high dimensionality of the latent space W^+, we first conduct principal component analysis (PCA) on the FFHQ dataset [Karras et al. 2020] mapped into W^+ using the image decoder. We obtain a subspace from the components with the largest eigenvalues, giving a basis $\mathbf{V}$. We project the identity latent w_I input into the subspace defined by $\mathbf{V}$ and concatenate it with the encoded audio segment e_t. This provides the input to the decoder. We map the decoder's output h_t from the subspace to W^+ to get the displacement vector d_t. Thus, we obtain $\bar{w}_t$ as follows:

$$\bar{w}_t = w_I + d_t = w_I + h_t \cdot \mathbf{V}, \quad t = 1, 2, 3, \cdots, T, \tag{4.3}$$

4.3.2 Training strategy

Our model is trained in two stages. In stage 1, we are only interested in learning trajectories in the latent space W^+ conditioned on the speech audio. The model predicts latent displacements to the identity image in the latent space of a fixed image generator. This disentangles the mouth

Layer type	Output	Activation
linear	512	ReLU
linear	512	ReLU
linear	2500	-

Table 4.4: The table describes the architecture of the latent decoder in the second model

motion and the facial identity. The motion trajectories are learned by training the model using a talking-head dataset. Although the model learns to generate accurate mouth movements, the visual quality of the generated video exhibits some distortion. The quality is determined by the pre-trained StyleGAN generator, which has been trained on images of people who are typically making a static pose. In stage 2, we tune the pre-trained StyleGAN image generator G on a single image or short video of a target speaker.

4.3.2.1 Stage 1: training the model

We train only the audio encoder E_A and latent decoder D while keeping the pre-trained image encoder E_I and the pre-trained image generator G fixed. For the loss function, we project the target frame x_t into the generator's latent space W^+ using the image encoder E_I. We then have the corresponding latent code $w_t = E_I(x_t)$ for $t = 1, 2, ..., T$. We calculate an L_2 loss between each target latent code w_t and the predicted latent code $\bar{w}_t$. We define $\mathcal{L}_{\text{latent}}$ as follow:

$$\mathcal{L}_{\text{latent}} = \sum_{t=1}^{T} ||w_t - \bar{w}_t||_2 \tag{4.4}$$

In addition, we apply another loss in the image domain between the generated video and the target video. Since the pre-trained image generator has not seen the training data, applying the loss directly on the target video would affect the model's performance, forcing it to focus on the facial appearance rather than mouth movements. For this, we project the target video using the image encoder and image generator $\hat{x_t} = G(E_I(x_t))$. We calculate the perceptual loss [R. Zhang et al. 2018] between the generated video and the projected target video as follows:

$$\mathcal{L}_{\text{LPIPS}} = \sum_{t=1}^{T} ||\phi(\bar{x_t}) - \phi(G(E_I(x_t)))||_2 \tag{4.5}$$

where ϕ is based on the VGG neural network [Simonyan and Zisserman 2015].

The overall loss for learning to predict latent displacements driven by audio is a weighted sum of the two losses:

$$\mathcal{L}_{\text{stage1}} = \lambda_{\text{latent}}\mathcal{L}_{\text{latent}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} \tag{4.6}$$

where λ_{latent} and λ_{latent} are hyperparameters to control the contribution of each corresponding loss.

4.3.2.2 Stage 2: tuning the image generator

One could tune both the image encoder E_I and the image generator G on the target speaker using an auto-encoder. However, this would transform the latent space W^+ and the learned model at stage 1 would consequently fail to generate correct mouth movements. To improve the visual quality of the generated video, we use the PTI method [Roich et al. 2021] to tune only the generator on a single image or video of a target speaker. In experiments, we implement stage 2 on short videos, and as a limiting case, on a single image.

Given a video of a speaker, we tune the image generator G on the constituent video frames x_t. For this task, we use the pre-trained encoder E_I to encode the frames x_t to the latent space W^+ to get w_t. Given $\hat{x}_t = G(w_t; \theta^*)$, we tune the weights of the generator while keeping the encoder fixed. We use the same objective loss used in PTI [Roich et al. 2021]:

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{LPIPS}} + \mathcal{L}_{\text{L}_2} \tag{4.7}$$

where $\mathcal{L}_{\text{L2}}$ is defined as :

$$\mathcal{L}_{\text{L}_2} = \sum_{t=1}^{T} ||(x_t - \hat{x}_t||_2 \tag{4.8}$$

After tuning the StyleGAN image generator, we can generate videos of talking heads using our inference pipeline with the components trained in both stages.

4.3.3 Dataset

We evaluate our approach using the two datasets we used in the first model (Section 4.2.3) for synthesizing talking-head videos: GRID [Cooke et al. 2006] and TCD-TIMIT [Harte and Gillen

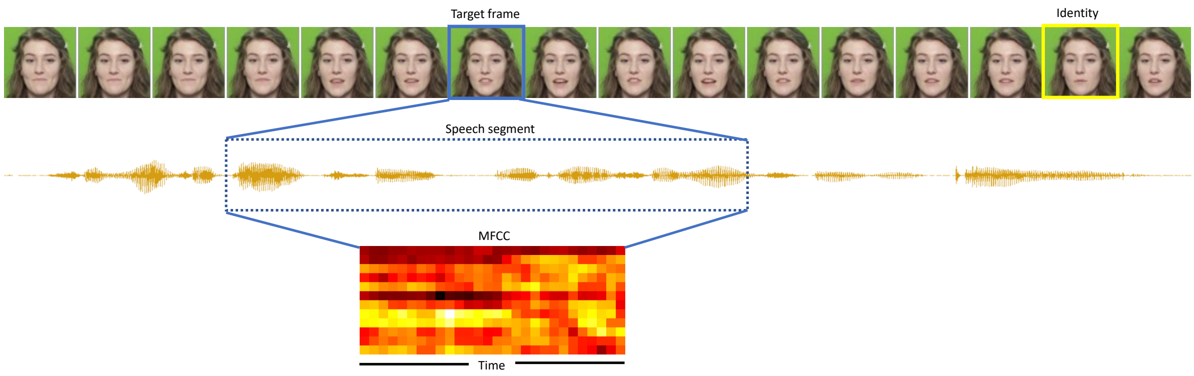


Figure 4.8: Sampling strategy of audio segments, identity images and the MFCC features in the second model. The identity images are randomly sampled from out of the audio sample window. The target frame is selected to be the middle frame of the audio window.

2015]. We hold out ten speakers from each dataset for testing and use the remaining speakers for training. For video pre-processing, we follow the process we discuss in Section 3.3.1.1. The input to the audio encoder E_A is an audio segment of length 280 ms which corresponds to a window of seven frames. However, we choose only the middle frame as the ground truth frame. The identity image is a randomly chosen frame out of this window.

Audio feature extraction. We represent the audio speech using MFCC values extracted from the raw values. MFCC is a widely used technique for extracting features in automatic speech recognition. These MFCC values consist of individual coefficients that correspond to specific frequency bands, measured on a non-linear mel scale. In our case, 13 coefficients are calculated per sample, but only the last 12 are used. Each sample provided to the audio encoder consists of 280 ms seconds of audio data sampled at 100 Hz, resulting in 28 time steps. The MFCC sample can be visualized as a 12×28 heatmap, with each column representing the MFCC coefficients at each time step. Figure 4.8 shows the sampling strategy of the identity image, the audio segment and the corresponding MFCC feature.

4.3.4 Implementation

We perform our experiments using PyTorch [Paszke et al. 2019]. For the StyleGAN image generator, we use a re-implementation of StyleGAN[2]. For the pre-trained image encoder, we use the original implementation of pSp [Richardson et al. 2021]. For training the audio encoder E_A and the decoder D in stage 1, we use an Adam optimiser Kingma and Ba 2014 with a learning rate of 0.0002. In Eqn. 4.6, we set $\lambda_{\text{latent}} = 250$ and $\lambda_{\text{LPIPS}} = 1$. We tune the

[2]https://github.com/rosinality/stylegan2-pytorch

generator with a learning rate of 0.0003. The tuning process takes less than two minutes for a single identity image. All experiments use an NVIDIA V100 GPU with 32 GB of memory.

4.3.5 Results

In this section, we evaluate our second model after tuning the generator in stage 2. Figure 4.9 shows videos generated using our second model. We use an unseen identity image and speech audio from two datasets: TCD-TIMIT and GRID. The figure demonstrates that our model can generate talking-head videos with accurate mouth movements matching the ground truth video. In addition, we analyse the model's capability to generate videos where the image generator is tuned on a single video frame and multiple video frames. Figure 4.10 shows the quality of the generated videos from tuning the generator on a single frame (middle row) and on a short video (bottom row). The figure indicates that tuning the generator on multiple frames results in better visual quality. This can be observed in the appearance of the eyes and mouth, which are enlarged and highlighted in red on the right side of the figure.

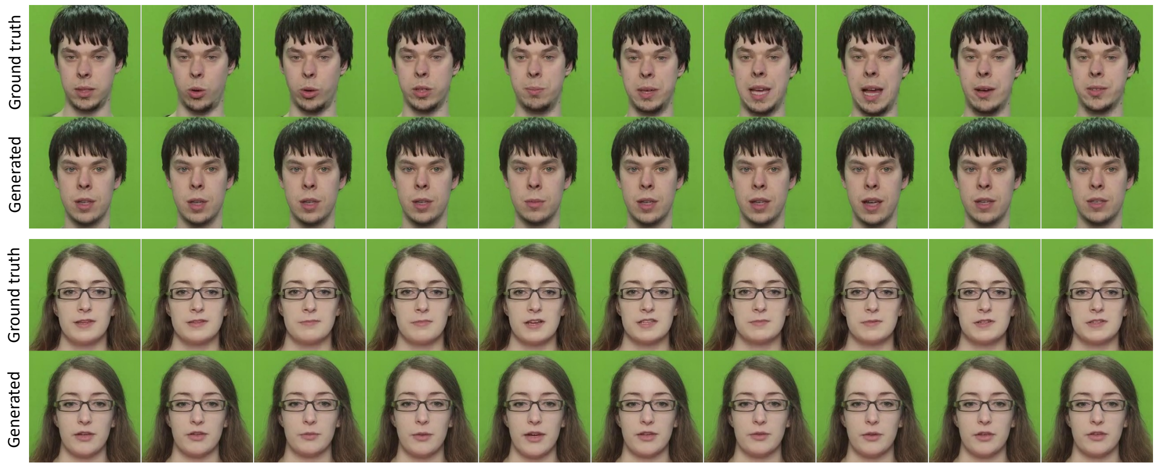
Ablation analysis. We present the results of the second model at Stage 1, without tuning the generator, and at Stage 2, after tuning the generator. Figure 4.11 shows videos generated using the trained model after Stage 1 and Stage 2. It is evident that the model at Stage 1 accurately generates mouth movements; however, the visual quality is compromised by some distortion introduced by the image generator. This distortion can be seen in the gaze of the eyes and the colour of the skin.

The appearance of the teeth in the Stage 2 results looks more realistic than in Stage 1, which is crucial for talking-head video generation. Therefore, Stage 2 is necessary to enhance the visual quality of the generated videos. In Stage 2, we tune the generator using either a single frame or a short video of the target speaker. The figure illustrates that Stage 2 does not compromise the performance of the videos generated from Stage 1.

4.4 Conclusion

In this chapter, we proposed an approach that generates talking-head videos from a given speech audio and identity image. The approach is built on top of a pre-trained StyleGAN image generator and image encoder. The idea is to project the identity image in the latent space and train a model that recurrently displaces the identity latent code conditioned on the

(a) Generated videos using testing videos from the GRID dataset



(b) Generated videos using testing videos from the TCD-TIMIT dataset

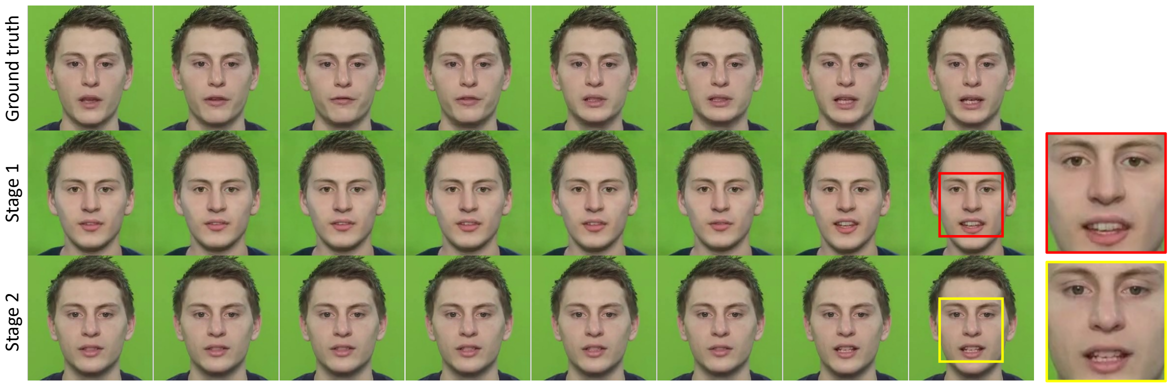
Figure 4.9: Generated videos of our second model. We generated the videos using the testing videos from two datasets: (a) GRID and (b) TCD-TIMIT. For each sub-figure, the first row represents the ground truth frames of the input speech audio while the second row shows the generated frames. From this figure, our second model synthesises videos well-synced with the input audio.

Figure 4.10: Samples generated using our approach. The top row shows frames from a source video providing the audio used to drive the generation. The middle row shows the corresponding generated frames where the generator G is tuned on a single frame. The bottom row shows generated frames where the generator G is tuned on a 5-second video clip. The figure demonstrates that tuning the generator with multiple frames leads to enhanced visual quality, as evident on the right side of the figure. These videos can be found at `https://mohammedalghamdi.github.io/talking-heads-acm-mm/`

speech audio. The image generator then synthesises the talking-head video from the sequence of displaced latent codes. For this approach, we proposed two models: the first model experiments with the StyleGAN pre-trained on video frames depicting talking heads (Section 4.2) and the second model experiments with the StyleGAN pre-trained on faces with static expressions (i.e., not talking) (Section 4.3). Both models were trained on talking-head datasets (TCD-TIMIT and GRID).

The first model generates realistic talking-head videos given novel speech audio but cannot generate videos from unseen given identities (seen in Figure 4.5). This is because the image encoder and image generator were pre-trained on talking-head video datasets of a few identities, around 50. The pre-trained StyleGAN image generator cannot synthesise new images of new identities beyond the ones depicted in the training videos (as discussed in Section 3.3.1.2). Ideally, the StyleGAN model requires to be trained on a large number of identities, i.e., approximately 70K to successfully generate images of new faces different from the training dataset. However, the first model can generate videos from unseen speech (Figure 4.4).

On the other hand, the second model can generate talking-head videos from unseen identities and speech audio. In contrast to the first model, the second model uses the MFCC feature to represent audio speech. In addition, the second model generates high-resolution videos (1024x1024 in our experiments). That is because the image generator was trained on high-resolution images. In the next chapter, we will evaluate the first and second models in comparison with other state-of-the-art models.



(a) Video samples generated from TCD-TIMIT dataset

(b) Video samples generated from GRID dataset

Figure 4.11: The figure displays videos generated by the second model after Stage 1 and Stage 2 from two datasets: (a) TCD-TIMIT and (b) GRID. It is evident from the figure that the results of Stage 1 produce accurate lip movements; however, there are some distortions in the eyes and the skin colour, which are highlighted in red. In Stage 2, the results appear more realistic than in Stage 1, particularly in the appearance of the teeth, which are highlighted in yellow.

Chapter 5

Evaluation

5.1 Introduction

In the preceding chapter, we built two generative models on top of a pre-trained StyleGAN image generator to synthesise realistic talking-head videos from speech audio. The first model uses a StyleGAN image generator pre-trained on frames of talking head videos while the second uses a StyleGAN pre-trained on the human face dataset.

In this chapter, we present quantitative and qualitative evaluations on both models in comparison with four state-of-the-art models: Vougioukas et al. 2018a, Chen et al. 2019a, Y. Zhou et al. 2020b and K R Prajwal et al. 2020a. We first introduce the quantitative evaluation measures used, followed by the evaluation results and discussion on two well-known talking-head video datasets: GRID [Cooke et al. 2006] and TCDTIMIT [Harte and Gillen 2015]. We also compare the computational complexity of the models in terms of inference time and size.

Secondly, we present the qualitative evaluation results in figures. This includes assessing the models in challenging scenarios such as bilabial events and identity images of speakers in the wild. Evaluating the models on bilabial events demonstrates how well these models can handle co-articulation problems. Co-articulation refers to the phenomenon where the articulation of one viseme is influenced by the adjacent visemes during visual speech production. Most of the current research papers on talking-head models do not evaluate the co-articulation issue; however, we believe this is a crucial step in assessing the naturalness of the generated videos.

In addition, we conduct a user study on the second model alongside two of the four state-of-the-art models. The user study is of great importance as it allows users to provide subjective feedback on the performance of the models. This includes the overall quality and lip-synchronisation of the generated video. The task of audio-driven talking-head video generation is one-to-many, meaning there are many possible correct ways of visually uttering the speech audio. For example, one can utter the same sentence with various non-lip facial expressions and correct lip movements. Most current quantitative measures do not address this issue; instead, they evaluate based on a single reference target video and penalise videos that generate different facial expressions than the target video. Therefore, a human user study is essential in addressing these issues.

Finally, we demonstrate that our models outperform the state-of-the-art models both quantitatively and qualitatively.

5.2 Quantitative evaluation

To evaluate talking-head models, we need to evaluate how well the lips synchronise with the given speech as well as the visual quality of generated frames. For the former, we use the landmarks distance measure (LMD) which computes the average Euclidean distance between the mouth landmarks of real video and generated video. For visual quality evaluation, we use the peak signal-to-noise ratio and the structural similarity (SSIM) which are common reconstruction measures. In addition, we use the Fréchet inception distance (FID) to measure how close the probability distribution of generated videos is to the real ones. For PSNR and SSIM, a higher score is better. For the FID and LMD, a lower score is better. In the following, we explain each measure in detail.

5.2.1 Measures

The Peak Signal-to-Noise Ratio (PSNR)

The peak signal-to-noise ratio is a function computed using the mean squared error (MSE) between the original image with the processed image. It represents the ratio between the maximum possible value of a signal and the MSE. Given a target video X and generated video $\bar{X}$, the PSNR is computed as follows:

$$PSNR(X, \bar{X}) = \frac{1}{T} \sum_{t=1}^{T} 10 \log_{10} \left(\frac{255^2}{MSE(x_t, \bar{x_t})} \right) \tag{5.1}$$

where x_t and $\bar{x_t}$ are the target video frame and generated video frame at time t. T is the number of video frames. 255 is the maximum intensity value of a pixel in a video frame. The MSE is computed as:

$$MSE(x, \bar{x}) = \frac{1}{CAB} \sum_{k=1}^{C} \sum_{i=1}^{A} \sum_{j=1}^{B} (x(k,i,j) - \bar{x}(k,i,j))^2 \tag{5.2}$$

where C is the number of colour channels. In our experiment, C is 3 to represent the Red, Green, and Blue colour channels. A and B represent the height and width of a video frame, respectively.

The Structural Similarity Index Measure (SSIM)

Z. Wang et al. 2004 introduce the SSIM measure, which is an alternative method that assesses the quality of an image by considering the degradation of its structural information. The pixels that are spatially close have inter-dependency that determines the visual structure of objects in a scene. The SSIM compares local pixel intensities after they have been normalised for luminance and contrast. The SSIM index is measured as follows:

$$SSIM(u, v) = \frac{(2\mu_u \mu_v + C_1)(2\sigma_u \sigma_v + C_2)}{({\mu_u}^2 + {\mu_v}^2 + C_1)({\sigma_u}^2 + {\sigma_v}^2 + C_2)} \tag{5.3}$$

where u and v are spatial patches extracted from a target frame x and a generated frame $\bar{x}$, respectively. μ_x and μ_v are the mean intensities computed over the pixels in each patch u and v, respectively. Similarly, σ_u and σ_v are the standard deviations of patch u and v, respectively. C_1 and C_2 are constants and set to 0.0001 and 0.0003, respectively. The measure is bounded above by 1, $SSIM(u, v) \leq 1$. To obtain a single SSIM value of videos, we calculate the mean of SSIM over the patches as the following:

$$MSSIM(X, \bar{X}) = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} SSIM(x_t(i), \bar{x}_t(i)) \tag{5.4}$$

where N is the number of patches in a single frame. In the thesis, we refer to MSSIM as SSIM in all our writings.

The Fréchet Inception Distance (FID)

Heusel et al. 2018 propose FID to quantitatively evaluate the quality of images generated using generative models, such as GANs. It computes the Fréchet distance between the real image distribution P_r and the generated image distribution P_g. It begins by representing images using features extracted from a pre-trained Inception model. The Inception model is trained to classify a large number of images and is known to capture meaningful and discriminative features from images. Next, the mean μ and covariance σ of these features are computed. The FID is then computed as follows:

$$FID(P_r, P_g) = ||\boldsymbol{\mu}_r - \boldsymbol{\mu}_g||_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}\right) \tag{5.5}$$

where $\boldsymbol{\mu}_r$ and $\boldsymbol{\mu}_g$ are the means of the real image features and generated image features. Σ_r and Σ_g are the covariances of the real image features and generated image features. Tr represents the trace of the matrix.

Using FID to measure the distance between target videos and generated videos, we first decode all generated videos and target videos of the testing set into video frames. We then follow the same steps described above to compute the statistics of the video frame features and the FID score. A lower score indicates that the generated videos are more similar to the target videos.

The Landmarks Distance Measure (LMD)

The landmarks distance measure (LMD) [Chen et al. 2018] is to evaluate the synchronisation between the mouth movement and the speech audio. This measure computes the Euclidean distance between the mouth landmarks of each generated frame and its corresponding true frame. It then averages the score on the number of frames and the number of mouth landmark points. For obtaining the mouth landmarks of a frame, we use the Face Alignment library Bulat and Tzimiropoulos 2017.

Suppose we have a target video X and a generated video $\bar{X}$, LMD is defined as

$$LMD(X, \bar{X}) = \frac{1}{TP} \sum_{t=1}^{T} \sum_{p=1}^{P} ||ML_X(p) - ML_{\bar{X}}(p)||_2 \tag{5.6}$$

Method	TCD-TIMIT				GRID			
	PSNR↑	SSIM↑	FID↓	LMD↓	PSNR↑	SSIM↑	FID↓	LMD↓
Vougioukas et al. 2018a	17.24	0.60	16.05	3.42	16.72	0.62	13.58	3.08
Chen et al. 2019a	15.31	0.58	11.79	3.66	16.80	<u>0.69</u>	13.27	3.74
Y. Zhou et al. 2020b	18.10	0.58	18.02	2.59	18.53	0.61	11.87	2.64
K R Prajwal et al. 2020a	18.26	0.64	15.24	2.19	17.83	<u>0.69</u>	11.11	2.05
Our first model	**21.14**	**0.65**	<u>8.11</u>	**1.76**	**21.84**	**0.73**	**5.30**	**1.66**
Our second model	<u>20.55</u>	**0.65**	**7.33**	<u>1.95</u>	<u>20.33</u>	0.65	<u>8.23</u>	<u>1.75</u>

Table 5.1: Quantitative comparisons of our first and second models with four state-of-the-art methods on two datasets (TCD-TIMIT and GRID). **First** and <u>second</u> best scores for each measure per dataset. ↑ indicates a higher score is better while ↓ indicates a lower score is better.

where ML is the mouth landmark, P is the number of ML points and T is the total number of frames in a video.

5.2.2 Results

We evaluate our two models introduced in the previous chapter against four state-of-the-art models Vougioukas et al. 2018b; Chen et al. 2019b; Y. Zhou et al. 2020a; K R Prajwal et al. 2020a using the aforementioned four measures. These models are the key competitors of our approach and aim to propose a generic method for synthesising talking-head videos given only speech audio and an identity image. Our approach differs from these models in several ways.

Firstly, our approach does not rely on learned intermediate features such as 2D facial landmarks, which are used in the works of Chen et al. 2019b and Y. Zhou et al. 2020a. These works map the audio to 2D facial landmarks and then from the landmarks to the video. Although facial landmarks facilitate the training of a talking-head model, they represent a sparse representation space, which may result in the loss of some information from the audio.

Secondly, unlike the approaches of Chen et al. 2019b and Vougioukas et al. 2018b, our approach does not generate a cropped face video. Instead, our approach, similar to the works of Y. Zhou et al. 2020a and K R Prajwal et al. 2020a, generates the full head.

Furthermore, our second model is capable of generating high-resolution videos from any identity image with considerably fewer learnable parameters than the competitor models. Scaling these models to generate high-resolution videos would significantly increase the training complexity and the number of parameters required. Instead, our approach is to utilise a pre-trained image

Method	Number of parameters	Inference time	Output size
Vougioukas et al. 2018a	55.28 M	441 FPS	96x128
Chen et al. 2019a	88.43 M	15.57 FPS	128x128
Y. Zhou et al. 2020b	36.40 M	10.32 FPS	256x256
K R Prajwal et al. 2020a	36.30 M	16 FPS	256x256
Our second model	29.68 M	35 FPS	1024x1024

Table 5.2: Comparisons between our second model and others in terms of the number of parameters, inference time, and output size.

generator that generates high-resolution images for generating videos thus reducing the learnable parameters and easing the training of the model.

We use the official codes of these models to generate the videos and compute the evaluation measures. We use the same testing videos to evaluate all models. None of the models had been exposed to the identities depicted in these testing videos during the training phase, except for our first model. However, our first model had not been trained on the speech audio from the testing videos. Table 5.1 shows that our first and second models significantly outperform other state-of-the-art models on the TCD-TIMIT dataset [Harte and Gillen 2015] and the GRID dataset [Cooke et al. 2006]. More importantly, our second model which is a generic model and can generate high-resolution videos outperforms the other four state-of-the-art methods on all measures except on the SSIM measure of the GRID dataset.

Computational complexity. We study the computational complexity of our second model in comparison to the four state-of-the-art models. In particular, we compute the number of trainable parameters of each model and the inference time each model takes to generate videos measured in frame per second (FPS). We also show the size of the output video frames of each model. Table 5.2 shows the number of trainable parameters, inference time and output frame size for the five methods. We ran all experiments on a V100 Nvidia GPU. The source videos used to compute the inference time are with a frame rate of 25. We can see that the method of Vougioukas et al. 2018b and our second model are faster than 25 FPS. In addition, our second model generates much higher-resolution videos compared to others.

5.3 Qualitative evaluation

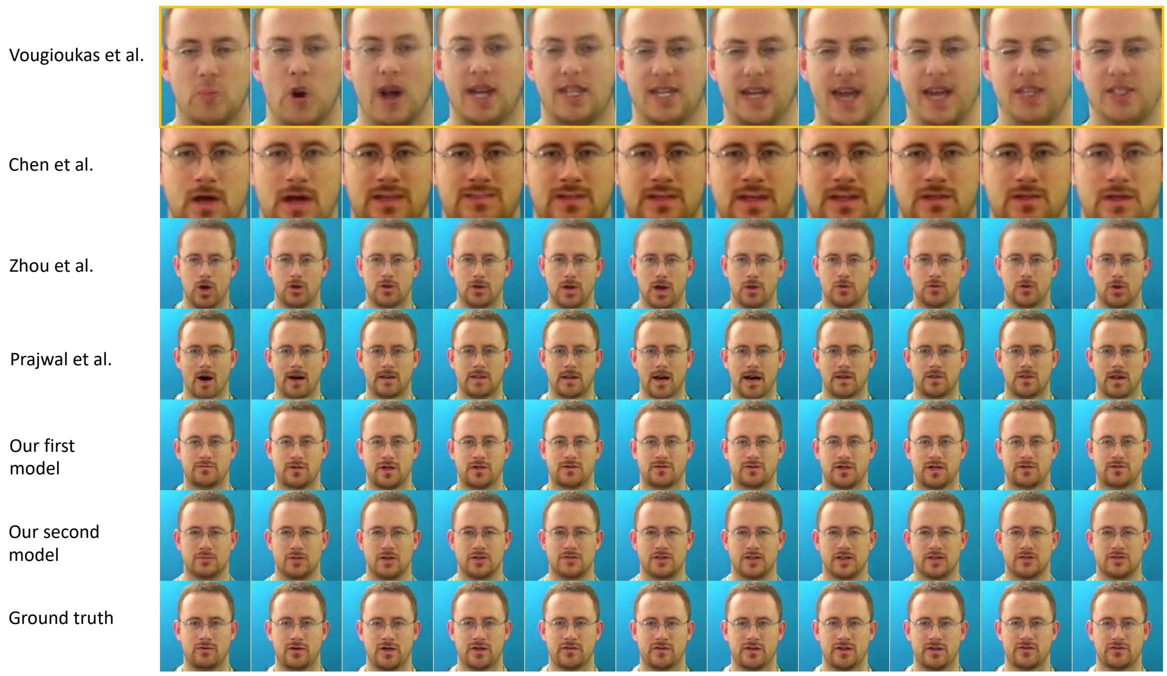
Here, we visualize the generated results of our models with the same four methods we compared against in the previous section. Figure 5.1 shows the visual quality of generated videos by our models in comparison with other models. The Sub-figure 5.1a shows that our two models generate photo-realistic videos largely indistinguishable from the ground truth (highlighted in yellow). We can see the mouth's shape of the work of K. R. Prajwal et al. 2020b is unrealistic to the ground truth (highlighted in blue). Our models preserve the identities throughout the frames in comparison with the work of Vougioukas et al. 2018b (highlighted in orange in the Sub-figure 5.1b).

Bilabial events. In the context of talking-head generation, bilabial events refer to lip movements during phonemes like /p/, /b/, and /m/. These phonemes require both lips to come into contact i.e., closed-mouth. This is crucial for generating photo-realistic and coherent talking-head videos. We study our two models in comparison with other state-of-the-art models during bilabial events. Figure 5.2 shows a visual comparison of our model with others on a challenging mouth movement associated with the phoneme /p/ in the word "place" (highlighted in yellow). It can be seen that our models and the work of Vougioukas et al. produce a closed-mouth shape in sync with the ground truth, while the work of Prajwal et al. is out of sync with the ground truth. In addition, the work of Chen et al. and Zhou et al. fail to produce the required mouth shape.

Unseen speech. We evaluate our models on unseen speech audio from the wild in comparison with the other works. This is to evaluate the lip-synchronisation quality and the robustness of a talking-head model to audios from out of the training distribution. For this, we extract the speech audio from a chosen video and test our models on the audio only while choosing an identity image from the training data. In this way, we can evaluate the abilities of the models to generalize on unseen voices. Figure 5.3 visualises the results of driving the video generation using unseen audio. The identity input image is from the GRID dataset. We can see that our second model generates accurate lip movements as the video of the source audio. Although our first model and other models generate reasonable mouth movements, they fail to produce an accurate mouth closure during some speech. This can be seen in the highlighted yellow frames.

Unseen identity. In addition, we also test our models on an unseen identity image from the wild while we choose speech audio from our training dataset. Figure 5.4 shows the visual result of testing the talking-head models on an unseen identity from the wild. We can see that our first model as expected did not preserve the identity because the pre-trained StyleGAN image generator was trained on a few speakers. However, the perceived identity is very close

(a) Video samples generated from the TCD-TIMIT dataset. The yellow-highlighted frames demonstrate that our models can generate frames that closely resemble the ground truth. The blue-highlighted frames show generated frames from the method of K. R. Prajwal et al. 2020b, in which the appearance of the mouth looks unrealistic.



(b) Video samples generated from the GRID dataset. The orange-highlighted frames show generated frames where the method of Vougioukas et al. 2018b fails to preserve the identity of the target speaker.

Figure 5.1: Qualitative comparisons: the figure shows video samples generated using our models in comparison with other state-of-the-art models on two benchmark datasets

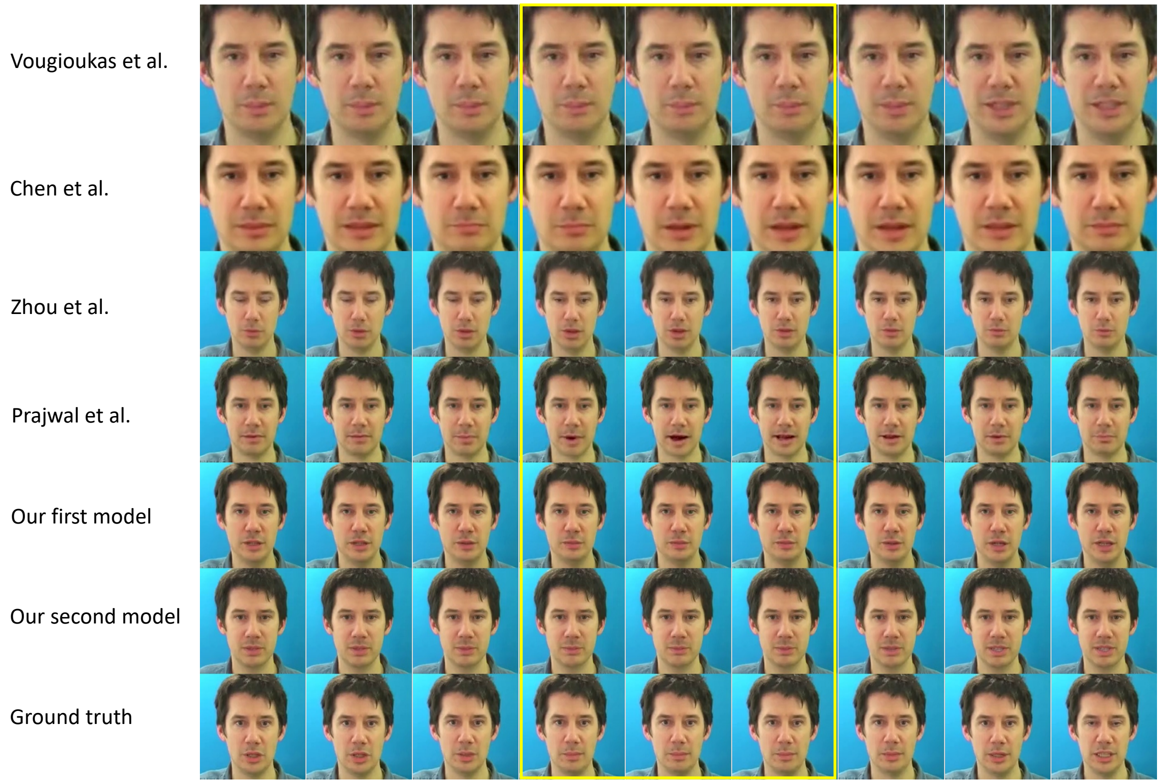


Figure 5.2: Visual comparison of mouth-closure during bilabial events during the /p/ phoneme (highlighted in yellow). We can see that our models and the work of Vougioukas et al. produce a closed-mouth shape while the work of Prajwal et al. is out of sync. In addition, the work of Chen et al. and Zhou et al. fail to produce the required mouth shape.

Figure 5.3: Unseen speech audio. The figure shows video samples generated with speech audio from the wild. It can be seen that our second model generates accurate mouth movements in comparison with others (highlighted in yellow)

Figure 5.4: Unseen identity image. The figure shows video samples generated with an identity (top row) from the wild. It can be seen that our first model as expected did not preserve the identity image because the pre-trained StyleGAN image generator was trained on a few speakers

to the training identities because of the learned identity encoder maps the given identity to the closest identity in the latent space. In addition, the identity remains unchanged during generated frames (highlighted in orange). The work of Vougioukas et al. 2018b fails to produce a plausible talking-head video. Our second model generates a photo-realistic talking-head video with accurate mouth movements. Both figures, 5.3 and 5.4, demonstrate the effectiveness of a talking-head model at disentangling the speech content from the identity content. This implies that the identity of the speaker is independent of the speech audio. This is very useful for applications such as video dubbing where the voice comes from another speaker.

5.4 User study

We conducted a user study to compare our second model with related works using Amazon Mechanical Turk services (MTurk). We evaluate both the audio-visual synchronisation and the visual quality of the state-of-the-art methods. We show participants a pair of videos: one generated by our method and the other generated by either Chen et al. 2019a or Y. Zhou et al. 2020b. For each pair, we either ask which video looks more photo-realistic or which video has

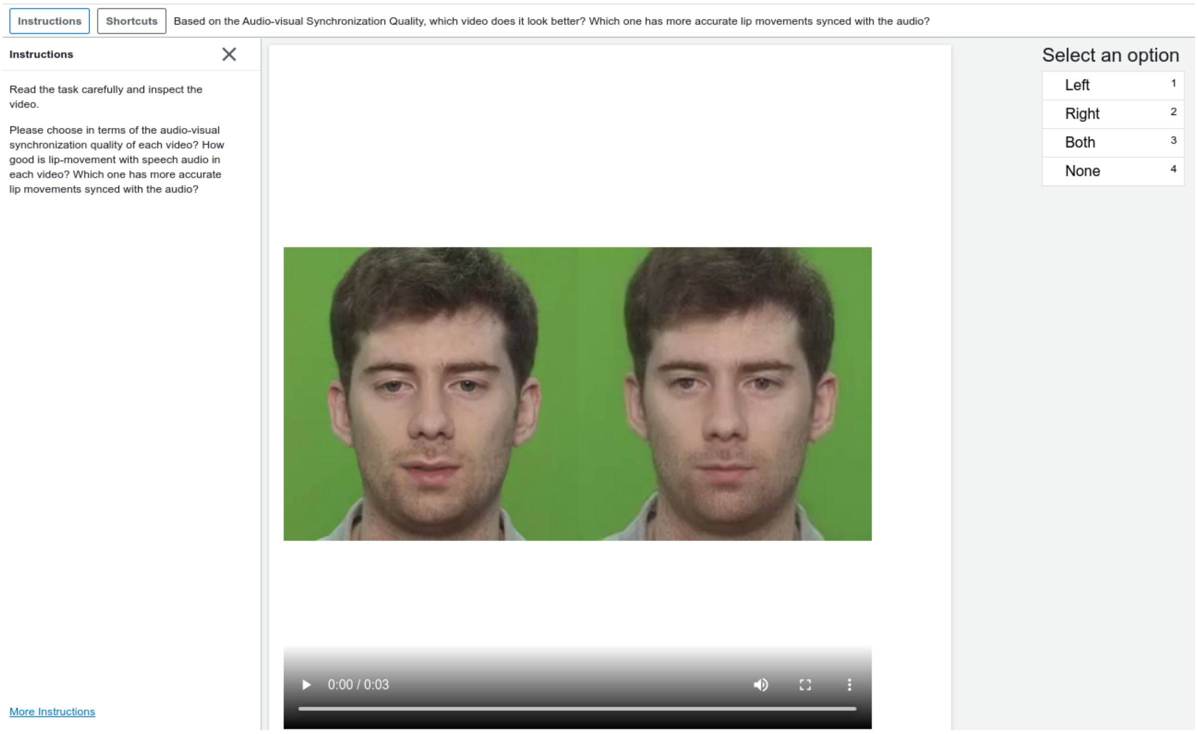


Figure 5.5: The interface of the user study shown to MTurk workers

a more accurate lip-sync with the audio. The two questions are:

- **The first question:** *Based on the visual quality of each video, which video does it look better? Which one is more photo-realistic than the other?*
- **The second question:** *Based on the Audio-visual Synchronization Quality, which video does it look better? Which one has more accurate lip movements synced with the audio?*

The choices for each question are "right", "left", "none", and "both". The "none" choice implies that none of the shown videos look good while the "both" choice implies that both shown videos look good. To ensure an unbiased presentation, we randomly determine the placement of videos in each pair, allowing our model-generated video to appear either on the left or the right. Figure 5.5 shows the interface of the questionnaire given to the participants. Before conducting the user study, we obtained ethics approval from the Faculty of Engineering and Physical Sciences (reference number: LTCOMP-004).

We conduct the user study on two datasets: GRID and TCD-TIMIT. We recruit 20 paid participants using MTurk services. The selection of participants are based on predefined settings set by us (the requester). For example, we only ask for participants from the United Kingdom since the generated videos are in English. We estimated the time it would take for each participant

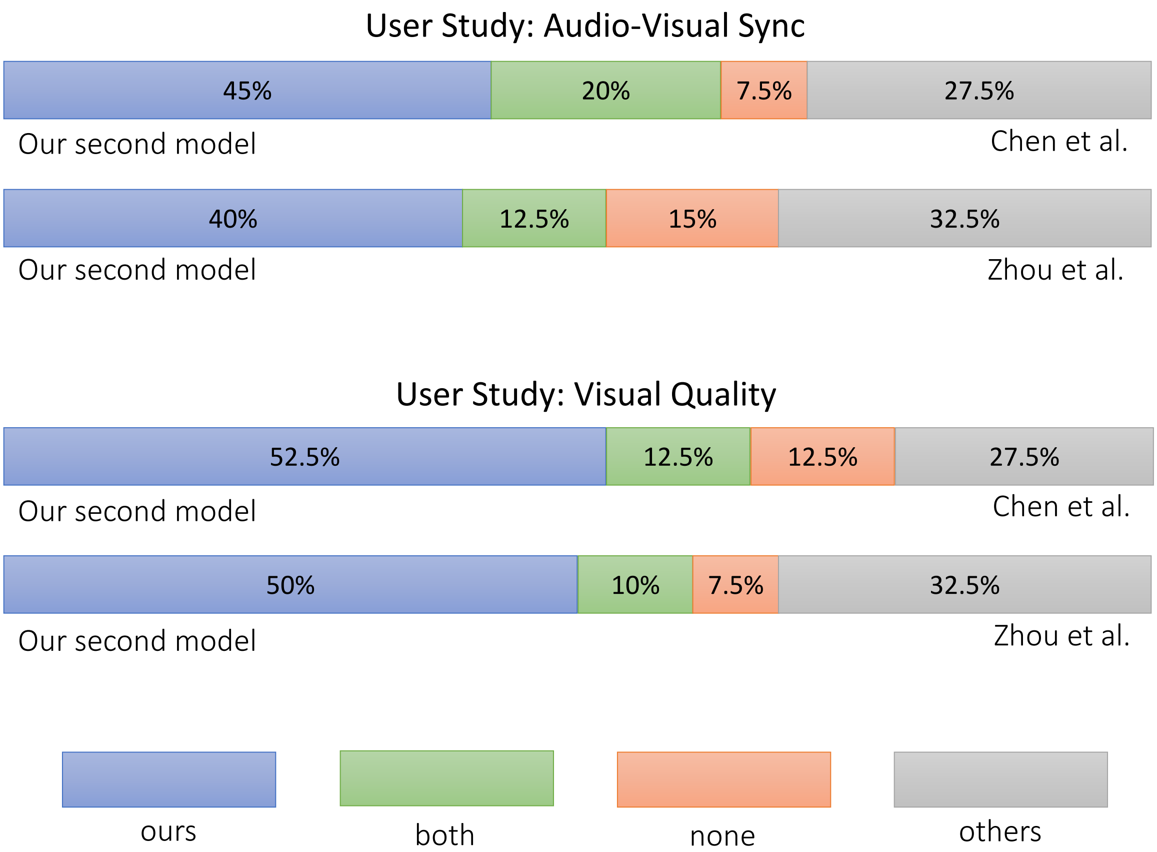


Figure 5.6: User study results for evaluating our second model with two state-of-the-art models in terms of the audio-visual synchronisation (top) and the visual quality (bottom)

to complete the evaluation. We anticipate that each participant would spend 30 seconds per 4-second video to answer a single question. We pay each participant more than double the National Living Wage of the UK, which amounted to £0.43 per answer. We obtain 80 answers from the 20 participants for each question. Figure 5.6 shows the results of the user study. It can be that seen our second model achieves a better result in terms of both the audio-visual synchronisation and the visual quality.

5.5 Conclusion and discussion

This chapter evaluates the performance of our first and second models in comparison to four state-of-the-art methods. We introduce common quantitative measures for evaluating talking-head videos, such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Fréchet Inception Distance (FID), and Landmark Distance (LMD). These measures assess visual quality and lip synchronization accuracy. We also compare the computational complexity of the models considering the number of parameters, inference time, and output size.

Qualitative evaluation results are presented in figures, highlighting the models' performance in challenging scenarios, including bilabial events, unseen speech audio, and identities "in the wild". Our evaluation is novel in that it specifically includes assessments of bilabial events, which is often overlooked in recent research papers. Furthermore, we present a user study comparing our second model with two state-of-the-art models.

The results show that our approach significantly outperforms the state-of-the-art models, both quantitatively and qualitatively. We attribute our models' superior performance to several factors. Regarding visual quality, our models leverage a pre-trained StyleGAN image generator, which is a leading-edge model in image synthesis. This powerful image generator is utilized to produce high-fidelity videos. Moreover, the StyleGAN model allows our approach to generalize across various identities since it was trained on extensive datasets of human faces Karras et al. 2019. Using a pre-trained image generator enables our approach to generate detailed articulations of the mouth for a given identity while the recurrent model can learn the mapping from audio into these detailed articulations with accurate synchronisation.

Chapter 6

Conclusion

In this thesis, we present a novel approach to synthesise a high-resolution talking-head video driven by speech audio. The task presents a challenge due to its requirement for expensive computation resources and the resulting increase in the complexity of the model. Our approach solves these issues by using a pre-trained StyleGAN image generator which is trained to synthesise high-resolution images of human faces. The approach is to train a speech-driven model that generates a sequence of latent code vectors in the latent space of the pre-rained StyleGAN image generator which then generates video frames of talking-head.

In Chapter 2, we start by reviewing relevant research in the area of audio-driven talking-head generation. We provide an overview of the early work for synthesising talking-head videos using learning-based and sample-based methods. Additionally, we review recent deep-learning methods including generic models and 3d facial animations. Furthermore, we explore existing talking-head video datasets. We also discuss GAN-based models for synthesising images including the StyleGAN model. We also delve into the discussion of GAN inversion methods, specifically encoder-based and optimisation-based, followed by understanding the latent space of a pre-trained StyleGAN model. Finally, we provide an overview of unconditional video generation methods, with a focus on the ones built on top of the StyleGAN model.

In Chapter 3, we investigate existing GAN inversion methods for projecting talking-head videos into the latent space of a pre-trained StyleGAN image generator. We modify an optimisation-based method for projecting videos into the latent space and show its results in reconstructing the input video frames efficiently. We also visualise the latent space with computed latent code vectors. In addition, we experiment with encoder-based methods for projecting video frames into the latent space. We show that encoder-based methods can project video frames effectively in a fraction of a second. Furthermore, we incorporate the encoder-based method for unconditional video generation. In particular, we propose a deep generative model that utilises the latent space of a pre-trained StyleGAN image generator with the learned projection to generate videos from a given noise vector. We run our experiments on a StyleGAN image generator pre-trained on video frames of talking heads and faces with static expressions.

In Chapter 4, we propose a novel approach using the learned projection to generate talking-head videos from speech audio. Our approach takes an identity image of the target speaker and encodes it using a pre-trained image encoder to find its latent code vector in the latent space of a pre-trained StyleGAN image generator. In addition to the identity image, the approach encodes a given speech audio using an audio encoder and trains a recurrent network that takes the encoded audio and generates a trajectory of latent code vectors. The StyleGAN generator takes these latent vectors to synthesise video frames of the target speaker uttering the speech. Finally, we show our approach can generate plausible high-quality videos with accurate lip movements.

In Chapter 5, we evaluate our approach presented in Chapter 4 for synthesising talking-head videos from speech audio. We use four quantitative evaluation measures (i.e., PSNR, SSIM, FID and LMD) to evaluate the lip-synchronisation and the visual quality of the generated video frames. We compare our approach with four state-of-the-art models on these four measures and show that our approach outperforms the others on all measures. In addition, we show the qualitative results of our approach with the other four models in different scenarios. This includes challenging scenarios during bilabial events. Furthermore, the chapter shows generated videos using speech audio and identity from the wild. Finally, we conduct a user study to compare our approach with two of the state-of-the-art methods.

6.1 Limitations

Although our approach generates accurate talking-head videos from speech audio, it has several limitations. First, the approach does not generate facial expressions that are not correlated to speech audio such as eye blinks. One reason is that the videos of our training datasets, GRID and TCD-TIMIT, are neutral and rarely exhibit any expression. It is possible that our approach could in principle generate other facial expressions if they are presented in the dataset, but we have not tested that yet. One practice in related works is to use an additional GAN loss such as a video discriminator trained on real and generated videos. This ensures that the model generates videos exhibiting natural movements similar to real videos. However, implementing this would increase the complexity of the model.

In addition, our approach cannot be trained on datasets of talking-head videos from the wild, like HDTF [Z. Zhang et al. 2021] or VoxCeleb [Nagrani et al. 2020] datasets, where the background of the videos is not blank. Video frames of talking heads need to be pre-processed, following the same alignment process used for pre-processing the training dataset (e.g., FFHQ) of the StyleGAN model [Karras et al. 2019]. The alignment process consists of applying a similarity transformation (scale, rotation, and translation) to match the faces with a reference face with a canonical position. Consequently, this leads to variations in the background, which therefore impacts the training of our approach since there is not just movement in the face but also in the background. Figure 6.1 shows the results of pre-processing videos from the wild using the same alignment method used in pre-processing our training datasets (Section 3.3.1.1). It can be observed that the background of the talking-head changes. One solution could be to pre-train the StyleGAN image generator on unaligned frames; however, we have tested this experiment and the image generator synthesises implausible images. Figure 6.2 shows image samples from the StyleGAN image generator trained on unaligned images. Another solution is to use the StyleGAN3 Karras et al. 2021 which is recently proposed to work on unaligned images of human faces. We plan to investigate this solution in the future.

6.2 Ethical consideration

Our framework can synthesise realistically high-quality videos from speech audio. This is perfect for video production animation, educational videos and video dubbing. For example, it can be used for generating personalised tutorial videos to improve specific skills of individual students. Creative people may use our work to edit content in movies or generate new videos.

In the recent past, significant advancements have been made in deep generative models, not just limited to video data but also images, audio, and text. Generated data using deep learning are becoming very hard to distinguish from real ones. They can be misused to spread misinformation or manipulate existing data. For example, they can generate fake news to manipulate public

Figure 6.1: This figure shows samples of talking-head videos that were collected from the wild. These videos were aligned using the same method that was used to pre-process the FFHQ dataset for the StyleGAN model [Section 3.3.1.1].

opinion in election campaigns. Therefore, this can lead to a loss of trust in digital information.

On the other hand, there have been promising advances in forensics in the detection of "deepfake" data [Rössler et al. 2018; Rössler et al. 2019; Cozzolino et al. 2018]. We will share generated videos from our work with the community to help detect manipulated videos. We hope these videos can contribute to the improvement of deepfake detectors. Any video generated using our approach will be clearly presented as synthetic.

6.3 Future work

We propose a novel approach for synthesising high-resolution videos from speech audio. The model can generate videos of a target speaker given a short video (or single image) of the speaker. However, the generated faces depict only mouth movements because the training datasets (TCD-TIMIT and GRID) are neutral and expressionless. We anticipate our approach could in principle generate other facial expressions where these are present in the dataset (e.g. K. Wang et al. 2020).

We train our approach with per-frame losses; we believe training the approach with a temporal

Figure 6.2: The figure demonstrates training the StyleGAN model on unaligned images results in unrealistic images. The upper sub-figure shows the real images of an unaligned training dataset used to train the StyleGAN model. The lower sub-figure shows generated images from the trained StyleGAN.

loss would enable the model to learn natural facial expressions. For example, a video discriminator network that is trained with GAN loss to take real video and generated video, and output a classification of real or fake. Another temporal loss would be an audio-visual loss to determine how accurate the lip movement is with the given speech audio [K R Prajwal et al. 2020a].

Our approach uses a pre-trained image encoder to encode each video frame into the latent space of a pre-trained StyleGAN image generator. The encoder does not take into account the temporal information in videos. One can benefit from developing a video encoder that encodes a sequence of video frames into the latent space. Such an encoder needs to disentangle constant features such as facial geometry, hair, and ears from changing features such as facial expression, mouth movements and head pose [Oorloff and Yacoob 2022]. We anticipate that incorporating such a video encoder in our approach might improve the performance.

Since our approach is built on top of the StyleGAN model [Karras et al. 2019], the video frames of our training dataset need to be aligned similarly to the alignment of the training dataset of StyleGAN. The recent follow-up work of StyleGAN3 Karras et al. 2021 is trained with an unaligned dataset and is better suited for videos and animation. Future work could be training our approach on the pre-trained StyleGAN3 image generator.

In addition, we believe there are still open problems in the task of audio-driven talking-head video generation. There are several recent works [Gan et al. 2023; Daněček et al. 2023; Peng et al. 2023] that propose emotional talking-head video generators; however, the overall quality of their results is still limited, and they are subject dependent, meaning they cannot generalize to any given identity. One possible reason could be the lack of available large emotional talking-head datasets. Most of the existing ones are lab-based datasets [K. Wang et al. 2020; Cao et al. 2014] and therefore are limited in the number of speakers (Section 2.2) as collecting a large dataset in the lab is very expensive. One solution is to collect emotional talking-head data from public websites such as YouTube.

Another open problem is generating a talking video of the full body rather than just the head. One of the challenges of this task is learning to generate hand gestures from speech audio. There are existing works [Bhattacharya et al. 2021; X. Liu et al. 2022] that generate hand gesture motion from speech audio. To the best of our knowledge, there are no existing works that aim at building a generative model to generate a video of the full body uttering the speech with plausible hand gestures.

Appendix A

Supplementary results for Chapter 3

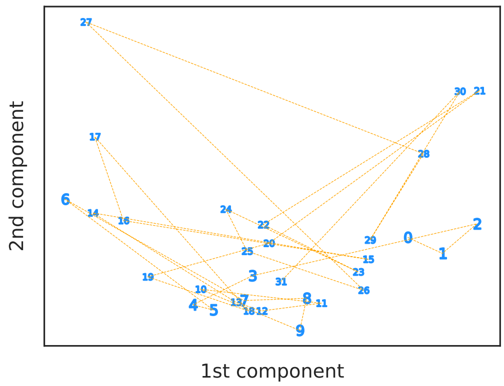
A.1 Optimisation-based GAN inversion

This appendix serves as a complementary extension to Chapter 3, providing additional results on the comparison between two optimisation-based projection algorithms, 1 and 2. Figures A.1 and A.2 show the results of using both algorithms to project video samples from the GRID and TCD-TIMIT datasets, respectively. It can be observed that Algorithm 2 produces very similar results to Algorithm 1, although the former is faster.

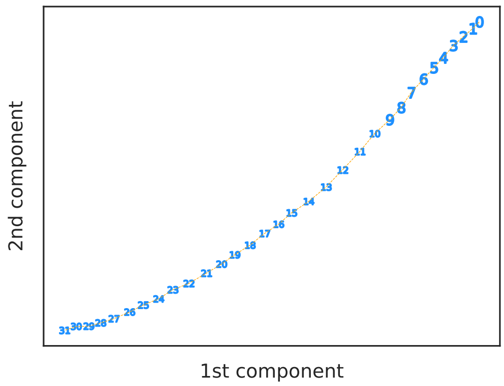
Figures A.3 and A.4 visualise the projection results in the latent space based on PCA computed per video. Algorithm 2 projects a video as a smooth trajectory in the latent space, whereas Algorithm 1 scatters the projections of a video.

Figure A.1: The middle row shows video frames projected into the W^+ latent space with the use of Algorithm 1, while the bottom row displays the results of using Algorithm 2. The original video frames, shown in the first row, come from the GRID dataset. Visually, it can be seen that the quality of the video frames projected using Algorithm 2 is comparable to those produced by Algorithm 1.

Figure A.2: The middle row shows video frames projected into the W^+ latent space with the use of Algorithm 1, while the bottom row displays the results of using Algorithm 2. The original video frames, shown in the first row, come from the TCD-TIMIT dataset. Visually, it can be seen that the quality of the video frames projected using Algorithm 2 is comparable to those produced by Algorithm 1.

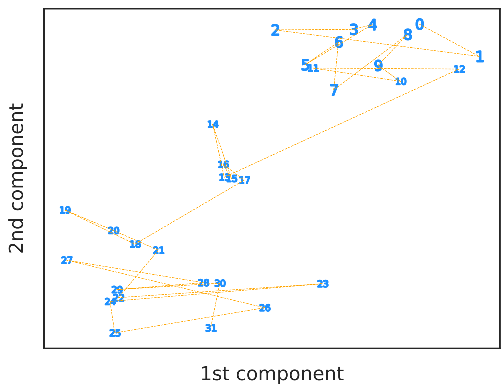


(a) Projection using Algorithm 1

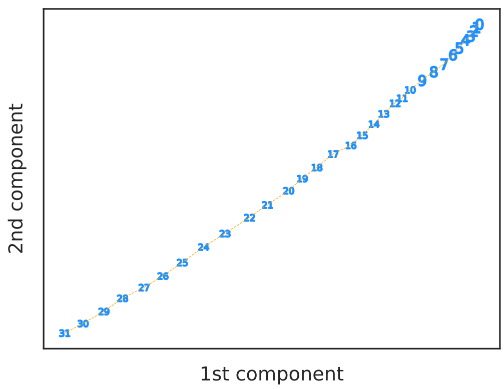


(b) Projection using Algorithm 2

Figure A.3: Visualisation of the video projection results using Algorithm 1 and Algorithm 2, based on the first two principal components computed for each video. Algorithm 2 projects a video as a smooth trajectory within the latent space. The video sample is taken from the GRID dataset.



(a) Projection using Algorithm 1



(b) Projection using Algorithm 2

Figure A.4: Visualisation of the video projection results using Algorithm 1 and Algorithm 2, based on the first two principal components computed for each video. Algorithm 2 projects a video as a smooth trajectory within the latent space. The video sample is taken from the TCD-TIMIT dataset.

References

Abdal, Rameen, Qin, Yipeng, and Wonka, Peter (2019a). "Image2stylegan: How to embed images into the stylegan latent space?" In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4432–4441.

Abdal, Rameen, Qin, Yipeng, and wonka, Peter (Sept. 2019b). "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?" In: *arXiv:1904.03189 [cs]*. arXiv: 1904.03189. URL: `http://arxiv.org/abs/1904.03189` (visited on 07/29/2020).

Abdal, Rameen, Zhu, Peihao, Mitra, Niloy, and Wonka, Peter (June 2021). "StyleFlow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows". In: *ACM Transactions on Graphics* 40.3. arXiv:2008.02401 [cs], pp. 1–21. ISSN: 0730-0301, 1557-7368. DOI: `10.1145/3447648`. URL: `http://arxiv.org/abs/2008.02401` (visited on 12/14/2023).

Acharya, Dinesh, Huang, Zhiwu, Paudel, Danda Pani, and Van Gool, Luc (Dec. 2018). *Towards High Resolution Video Generation with Progressive Growing of Sliced Wasserstein GANs*. arXiv:1810.02419 [cs, stat]. DOI: `10.48550/arXiv.1810.02419`. URL: `http://arxiv.org/abs/1810.02419` (visited on 05/10/2023).

Afouras, Triantafyllos, Chung, Joon Son, Senior, Andrew W., Vinyals, Oriol, and Zisserman, Andrew (2018). "Deep Audio-Visual Speech Recognition". In: *CoRR* abs/1809.02108. arXiv: `1809.02108`. URL: `http://arxiv.org/abs/1809.02108`.

Alaluf, Yuval, Patashnik, Or, and Cohen-Or, Daniel (Aug. 2021). *ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement*. arXiv:2104.02699 [cs]. DOI: `10.48550/arXiv.2104.02699`. URL: `http://arxiv.org/abs/2104.02699` (visited on 11/07/2022).

Alaluf, Yuval, Tov, Omer, Mokady, Ron, Gal, Rinon, and Bermano, Amit H. (Mar. 2022). *HyperStyle: StyleGAN Inversion with HyperNetworks for Real Image Editing*. Tech. rep. arXiv:2111.15666. arXiv:2111.15666 [cs] type: article. arXiv. URL: `http://arxiv.org/abs/2111.15666` (visited on 05/19/2022).

Alghamdi, Mohammed M., Wang, He, Bulpitt, Andrew J., and Hogg, David C. (Oct. 2022). "Talking Head from Speech Audio using a Pre-trained Image Generator". In: *Proceedings of the 30th ACM International Conference on Multimedia*. arXiv:2209.04252 [cs], pp. 5228–5236. DOI: `10.1145/3503161.3548101`. URL: `http://arxiv.org/abs/2209.04252` (visited on 05/12/2023).

Bau, David, Zhu, Jun-Yan, Wulff, Jonas, Peebles, William, Strobelt, Hendrik, Zhou, Bolei, and Torralba, Antonio (Oct. 2019). "Seeing What a GAN Cannot Generate". en. In: *arXiv:1910.11626 [cs, eess]*. arXiv: 1910.11626. URL: `http://arxiv.org/abs/1910.11626` (visited on 05/10/2022).

Bhattacharya, Uttaran, Childs, Elizabeth, Rewkowski, Nicholas, and Manocha, Dinesh (Oct. 2021). "Speech2AffectiveGestures: Synthesizing Co-Speech Gestures with Generative Adversarial Affective Expression Learning". en. In: *Proceedings of the 29th ACM International Conference on Multimedia*. Virtual Event China: ACM, pp. 2027–2036. ISBN: 978-1-4503-8651-7. DOI: `10.1145/3474085.3475223`. URL: `https://dl.acm.org/doi/10.1145/3474085.3475223` (visited on 12/10/2021).

Brand, Matthew (1999). "Voice puppetry". In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 21–28.

Bregler, Christoph, Covell, Michele, and Slaney, Malcolm (1997). "Video rewrite: Driving visual speech with audio". In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 353–360.

Brock, Andrew, Donahue, Jeff, and Simonyan, Karen (Feb. 2019). "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *arXiv:1809.11096 [cs, stat]*. arXiv: 1809.11096. URL: `http://arxiv.org/abs/1809.11096` (visited on 08/08/2020).

Bulat, Adrian and Tzimiropoulos, Georgios (Oct. 2017). “How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks)”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. arXiv:1703.07332 [cs], pp. 1021–1030. DOI: 10.1109/ICCV.2017.116. URL: http://arxiv.org/abs/1703.07332 (visited on 06/15/2023).

Cao, Houwei, Cooper, David G, Keutmann, Michael K, Gur, Ruben C, Nenkova, Ani, and Verma, Ragini (2014). “Crema-d: Crowd-sourced emotional multimodal actors dataset”. In: *IEEE transactions on affective computing* 5.4, pp. 377–390.

Charles, James, Magee, Derek, and Hogg, David (2016). “Virtual immortality: Reanimating characters from tv shows”. In: *European Conference on Computer Vision*. Springer, pp. 879–886.

Chen, Lele, Cui, Guofeng, Kou, Ziyi, Zheng, Haitian, and Xu, Chenliang (May 2020). “What comprises a good talking-head video generation?: A Survey and Benchmark”. In: *arXiv:2005.03201 [cs, eess]*. arXiv: 2005.03201. URL: http://arxiv.org/abs/2005.03201 (visited on 06/01/2020).

Chen, Lele, Li, Zhiheng, Maddox, Ross K., Duan, Zhiyao, and Xu, Chenliang (2018). “Lip Movements Generation at a Glance”. en. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Vol. 11211. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 538–553. ISBN: 978-3-030-01233-5 978-3-030-01234-2. DOI: 10.1007/978-3-030-01234-2_32. URL: http://link.springer.com/10.1007/978-3-030-01234-2_32 (visited on 06/15/2021).

Chen, Lele, Maddox, Ross K, Duan, Zhiyao, and Xu, Chenliang (2019a). “Hierarchical cross-modal talking face generation with dynamic pixel-wise loss”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7832–7841.

– (2019b). “Hierarchical cross-modal talking face generation with dynamic pixel-wise loss”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7832–7841.

Choi, Kyoungho, Luo, Ying, and Hwang, Jenq-Neng (Aug. 2001a). “Hidden Markov Model Inversion for Audio-to-Visual Conversion in an MPEG-4 Facial Animation System”. en. In: *Journal of VLSI signal processing systems for signal, image and video technology* 29.1, pp. 51–61. ISSN: 0922-5773. DOI: 10.1023/A:1011171430700. URL: https://doi.org/10.1023/A:1011171430700 (visited on 06/27/2022).

– (2001b). “Hidden Markov model inversion for audio-to-visual conversion in an MPEG-4 facial animation system”. In: *Journal of VLSI signal processing systems for signal, image and video technology* 29.1, pp. 51–61.

Chung, Joon Son, Jamaludin, Amir, and Zisserman, Andrew (July 2017a). “You said that?” In: *arXiv:1705.02966 [cs]*. arXiv: 1705.02966. URL: http://arxiv.org/abs/1705.02966 (visited on 05/09/2022).

– (2017b). “You said that?” In: *British Machine Vision Conference*.

Chung, Joon Son, Nagrani, Arsha, and Zisserman, Andrew (Sept. 2018). “VoxCeleb2: Deep Speaker Recognition”. en. In: *Interspeech 2018*. ISCA, pp. 1086–1090. DOI: 10.21437/Interspeech.2018-1929. URL: https://www.isca-speech.org/archive/interspeech_2018/chung18b_interspeech.html (visited on 08/03/2022).

Cooke, Martin, Barker, Jon, Cunningham, Stuart P., and Shao, Xu (2006). “An audio-visual corpus for speech perception and automatic speech recognition.” In: *The Journal of the Acoustical Society of America* 120 5 Pt 1, pp. 2421–4.

Cozzolino, Davide, Thies, Justus, Rössler, Andreas, Riess, Christian, Nießner, Matthias, and Verdoliva, Luisa (2018). “ForensicTransfer: Weakly-supervised Domain Adaptation for Forgery Detection”. In: *arXiv*.

Cudeiro, Daniel, Bolkart, Timo, Laidlaw, Cassidy, Ranjan, Anurag, and Black, Michael (2019). “Capture, Learning, and Synthesis of 3D Speaking Styles”. In: *Computer Vision and Pattern Recognition (CVPR)*.

Daněček, Radek, Chhatre, Kiran, Tripathi, Shashank, Wen, Yandong, Black, Michael J., and Bolkart, Timo (June 2023). *Emotional Speech-Driven Animation with Content-Emotion Dis-*

entanglement. arXiv:2306.08990 [cs]. URL: http://arxiv.org/abs/2306.08990 (visited on 06/22/2023).

Das, Dipanjan, Biswas, S., Sinha, Sanjana, and Bhowmick, Brojeshwar (2020). "Speech-Driven Facial Animation Using Cascaded GANs for Learning of Motion and Texture". In: *ECCV*.

Deng, Jiankang, Guo, Jia, Yang, Jing, Xue, Niannan, Kotsia, Irene, and Zafeiriou, Stefanos (2021). "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. arXiv:1801.07698 [cs], pp. 1–1. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2021.3087709. URL: http://arxiv.org/abs/1801.07698 (visited on 12/20/2022).

Deng, Zhigang and Noh, Junyong (2008). "Computer facial animation: A survey". In: *Data-driven 3D facial animation*. Springer, pp. 1–28.

Ezzat, Tony, Geiger, Gadi, and Poggio, Tomaso (2002). "Trainable videorealistic speech animation". In: *ACM Transactions on Graphics (TOG)* 21.3, pp. 388–398.

Fan, Bo, Wang, Lijuan, Soong, Frank K., and Xie, Lei (Apr. 2015). "Photo-real talking head with deep bidirectional LSTM". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ISSN: 2379-190X, pp. 4884–4888. DOI: 10.1109/ICASSP.2015.7178899.

Fan, Yingruo, Lin, Zhaojiang, Saito, Jun, Wang, Wenping, and Komura, Taku (Mar. 2022). *FaceFormer: Speech-Driven 3D Facial Animation with Transformers*. arXiv:2112.05329 [cs]. URL: http://arxiv.org/abs/2112.05329 (visited on 05/10/2023).

Forney, G David (1973). "The viterbi algorithm". In: *Proceedings of the IEEE* 61.3, pp. 268–278.

Fox, Gereon, Tewari, Ayush, Elgharib, Mohamed, and Theobalt, Christian (2021). "StyleVideoGAN: A Temporal Generative Model using a Pretrained StyleGAN". In: URL: https://arxiv.org/pdf/2107.07224.

Fu, Shengli, Gutierrez-Osuna, R., Esposito, A., Kakumanu, P.K., and Garcia, O.N. (2005). "Audio/visual mapping with cross-modal hidden Markov models". In: *IEEE Transactions on Multimedia* 7.2, pp. 243–252. DOI: 10.1109/TMM.2005.843341.

Gan, Yuan, Yang, Zongxin, Yue, Xihang, Sun, Lingyun, and Yang, Yi (Oct. 2023). *Efficient Emotional Adaptation for Audio-Driven Talking-Head Generation.* arXiv:2309.04946 [cs, eess]. DOI: 10.48550/arXiv.2309.04946. URL: http://arxiv.org/abs/2309.04946 (visited on 12/18/2023).

Garofolo, John S (1993). "Timit acoustic phonetic continuous speech corpus". In: *Linguistic Data Consortium, 1993.*

Garrido, Pablo, Valgaerts, Levi, Sarmadi, Hamid, Steiner, Ingmar, Varanasi, Kiran, Pérez, Patrick, and Theobalt, Christian (2015). "VDub: Modifying Face Video of Actors for Plausible Visual Alignment to a Dubbed Audio Track". In: *Computer Graphics Forum* 34.

Goodfellow, Ian (2016). "Nips 2016 tutorial: Generative adversarial networks". In: *arXiv preprint arXiv:1701.00160.*

– (Apr. 2017). *NIPS 2016 Tutorial: Generative Adversarial Networks.* arXiv:1701.00160 [cs]. DOI: 10.48550/arXiv.1701.00160. URL: http://arxiv.org/abs/1701.00160 (visited on 06/26/2023).

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems.* Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger. Vol. 27. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Hannun, Awni Y., Case, Carl, Casper, J., Catanzaro, Bryan, Diamos, G., Elsen, Erich, Prenger, R., Satheesh, S., Sengupta, Shubho, Coates, A., and Ng, A. (2014). "Deep Speech: Scaling up end-to-end speech recognition". In: *ArXiv* abs/1412.5567.

Härkönen, Erik, Hertzmann, Aaron, Lehtinen, Jaakko, and Paris, Sylvain (2020). "Ganspace: Discovering interpretable gan controls". In: *Advances in Neural Information Processing Systems* 33, pp. 9841–9850.

Harte, Naomi and Gillen, Eoin (2015). "TCD-TIMIT: An Audio-Visual Corpus of Continuous Speech". In: *IEEE Transactions on Multimedia* 17, pp. 603–615.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian (2016). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp (Jan. 2018). *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium.* arXiv:1706.08500 [cs, stat]. DOI: 10.48550/arXiv.1706.08500. URL: http://arxiv.org/abs/1706.08500 (visited on 12/06/2022).

Isola, Phillip, Zhu, Jun-Yan, Zhou, Tinghui, and Efros, Alexei A. (2017). "Image-to-Image Translation with Conditional Adversarial Networks". In: *Proc. CVPR*, pp. 5967–5976. DOI: 10.1109/CVPR.2017.632.

Ji, Xinya, Zhou, Hang, Wang, Kaisiyuan, Wu, Wayne, Loy, Chen Change, Cao, Xun, and Xu, Feng (May 2021). "Audio-Driven Emotional Video Portraits". In: *arXiv:2104.07452 [cs].* arXiv: 2104.07452. URL: http://arxiv.org/abs/2104.07452 (visited on 05/24/2021).

Johnson, Justin, Alahi, Alexandre, and Fei-Fei, Li (2016). "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: *ECCV*, pp. 694–711.

Karras, Tero, Aila, Timo, Laine, Samuli, Herva, Antti, and Lehtinen, Jaakko (Aug. 2017a). "Audio-driven facial animation by joint end-to-end learning of pose and emotion". en. In: *ACM Transactions on Graphics* 36.4, pp. 1–12. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3072959.3073658. URL: https://dl.acm.org/doi/10.1145/3072959.3073658 (visited on 03/29/2023).

Karras, Tero, Aila, Timo, Laine, Samuli, and Lehtinen, Jaakko (2017b). "Progressive growing of gans for improved quality, stability, and variation". In: *arXiv preprint arXiv:1710.10196.*

– (2018). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *Proc. ICLR.*

Karras, Tero, Aittala, Miika, Laine, Samuli, Härkönen, Erik, Hellsten, Janne, Lehtinen, Jaakko, and Aila, Timo (Oct. 2021). "Alias-Free Generative Adversarial Networks". In: *arXiv:2106.12423 [cs, stat].* arXiv: 2106.12423. URL: http://arxiv.org/abs/2106.12423 (visited on 12/10/2021).

Karras, Tero, Laine, Samuli, and Aila, Timo (Mar. 2019). “A Style-Based Generator Architecture for Generative Adversarial Networks”. In: *arXiv:1812.04948 [cs, stat]*. arXiv: 1812.04948. URL: `http://arxiv.org/abs/1812.04948` (visited on 11/20/2020).

Karras, Tero, Laine, Samuli, Aittala, Miika, Hellsten, Janne, Lehtinen, Jaakko, and Aila, Timo (2020). “Analyzing and improving the image quality of stylegan”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119.

Kingma, Diederik P and Ba, Jimmy (2014). “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980*.

Kingma, Diederik P and Welling, Max (2013). “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114*.

Lin, Tsung-Yi, Dollár, Piotr, Girshick, Ross, He, Kaiming, Hariharan, Bharath, and Belongie, Serge (Apr. 2017). *Feature Pyramid Networks for Object Detection*. arXiv:1612.03144 [cs]. DOI: `10.48550/arXiv.1612.03144`. URL: `http://arxiv.org/abs/1612.03144` (visited on 12/06/2022).

Liu, Xian, Wu, Qianyi, Zhou, Hang, Du, Yuanqi, Wu, Wayne, Lin, Dahua, and Liu, Ziwei (Dec. 2022). *Audio-Driven Co-Speech Gesture Video Generation*. arXiv:2212.02350 [cs] version: 1. DOI: `10.48550/arXiv.2212.02350`. URL: `http://arxiv.org/abs/2212.02350` (visited on 12/07/2022).

Nagrani, Arsha, Chung, Joon Son, Xie, Weidi, and Zisserman, Andrew (2020). “Voxceleb: Large-scale speaker verification in the wild”. In: *Computer Speech & Language* 60, p. 101027.

Oorloff, Trevine and Yacoob, Yaser (Mar. 2022). *Encode-in-Style: Latent-based Video Encoding using StyleGAN2*. Tech. rep. arXiv:2203.14512. arXiv:2203.14512 [cs] type: article. arXiv. URL: `http://arxiv.org/abs/2203.14512` (visited on 05/19/2022).

Parke, Frederic I. and Waters, Keith (Sept. 2008). *Computer Facial Animation*. en. Google-Books-ID: yEzrBgAAQBAJ. CRC Press. ISBN: 978-1-4398-6540-8.

Paszke, Adam, Gross, Sam, Massa, Francisco, Lerer, Adam, Bradbury, James, Chanan, Gregory, Killeen, Trevor, Lin, Zeming, Gimelshein, Natalia, Antiga, Luca, Desmaison, Alban, Kopf, Andreas, Yang, Edward, DeVito, Zachary, Raison, Martin, Tejani, Alykhan, Chilamkurthy, Sasank, Steiner, Benoit, Fang, Lu, Bai, Junjie, and Chintala, Soumith (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Peng, Ziqiao, Wu, Haoyu, Song, Zhenbo, Xu, Hao, Zhu, Xiangyu, He, Jun, Liu, Hongyan, and Fan, Zhaoxin (Aug. 2023). *EmoTalk: Speech-Driven Emotional Disentanglement for 3D Face Animation.* arXiv:2303.11089 [cs, eess]. DOI: `10.48550/arXiv.2303.11089`. URL: `http://arxiv.org/abs/2303.11089` (visited on 12/18/2023).

Prajwal, K R, Mukhopadhyay, Rudrabha, Namboodiri, Vinay P., and Jawahar, C.V. (2020a). "A Lip Sync Expert Is All You Need for Speech to Lip Generation In the Wild". In: *Proceedings of the 28th ACM International Conference on Multimedia.* MM '20. Seattle, WA, USA: Association for Computing Machinery, pp. 484–492. ISBN: 9781450379885. DOI: `10.1145/3394171.3413532`. URL: `https://doi.org/10.1145/3394171.3413532`.

Prajwal, K. R., Mukhopadhyay, Rudrabha, Namboodiri, Vinay, and Jawahar, C. V. (Oct. 2020b). "A Lip Sync Expert Is All You Need for Speech to Lip Generation In The Wild". In: *Proceedings of the 28th ACM International Conference on Multimedia.* arXiv: 2008.10010, pp. 484–492. DOI: `10.1145/3394171.3413532`. URL: `http://arxiv.org/abs/2008.10010` (visited on 04/08/2022).

Rabiner, Lawrence R (1989). "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286.

Radford, Alec, Metz, Luke, and Chintala, Soumith (2016). "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *Proc. ICLR.*

Richard, Alexander, Zollhoefer, Michael, Wen, Yandong, Torre, Fernando de la, and Sheikh, Yaser (Apr. 2021a). "MeshTalk: 3D Face Animation from Speech using Cross-Modality Disentanglement". In: *arXiv:2104.08223 [cs].* arXiv: 2104.08223. URL: `http://arxiv.org/abs/2104.08223` (visited on 04/19/2021).

Richard, Alexander, Zollhoefer, Michael, Wen, Yandong, Torre, Fernando de la, and Sheikh, Yaser (2021b). *MeshTalk: 3D Face Animation from Speech using Cross-Modality Disentanglement*. arXiv: `2104.08223 [cs.CV]`.

Richardson, Elad, Alaluf, Yuval, Patashnik, Or, Nitzan, Yotam, Azar, Yaniv, Shapiro, Stav, and Cohen-Or, Daniel (Apr. 2021). "Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation". In: *arXiv:2008.00951 [cs]*. arXiv: 2008.00951. URL: `http://arxiv.org/abs/2008.00951` (visited on 03/04/2022).

Roich, Daniel, Mokady, Ron, Bermano, Amit H, and Cohen-Or, Daniel (2021). "Pivotal tuning for latent-based editing of real images". In: *arXiv preprint arXiv:2106.05744*.

Rössler, Andreas, Cozzolino, Davide, Verdoliva, Luisa, Riess, Christian, Thies, Justus, and Nießner, Matthias (2018). "FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces". In: *arXiv*.

– (Aug. 2019). *FaceForensics++: Learning to Detect Manipulated Facial Images*. arXiv:1901.08971 [cs]. DOI: `10.48550/arXiv.1901.08971`. URL: `http://arxiv.org/abs/1901.08971` (visited on 05/12/2023).

Saito, Masaki, Matsumoto, Eiichi, and Saito, Shunta (Aug. 2017). *Temporal Generative Adversarial Nets with Singular Value Clipping*. arXiv:1611.06624 [cs]. DOI: `10.48550/arXiv.1611.06624`. URL: `http://arxiv.org/abs/1611.06624` (visited on 05/10/2023).

Saito, Masaki, Saito, Shunta, Koyama, Masanori, and Kobayashi, Sosuke (Nov. 2020). "Train Sparsely, Generate Densely: Memory-efficient Unsupervised Training of High-resolution Temporal GAN". In: *International Journal of Computer Vision* 128.10-11. arXiv:1811.09245 [cs], pp. 2586–2606. ISSN: 0920-5691, 1573-1405. DOI: `10.1007/s11263-020-01333-y`. URL: `http://arxiv.org/abs/1811.09245` (visited on 05/10/2023).

Shen, Shuai, Zhao, Wenliang, Meng, Zibin, Li, Wanhua, Zhu, Zheng, Zhou, Jie, and Lu, Jiwen (Apr. 2023). *DiffTalk: Crafting Diffusion Models for Generalized Audio-Driven Portraits Animation*. arXiv:2301.03786 [cs]. DOI: `10.48550/arXiv.2301.03786`. URL: `http://arxiv.org/abs/2301.03786` (visited on 12/13/2023).

Shen, Yujun, Gu, Jinjin, Tang, Xiaoou, and Zhou, Bolei (2020). "Interpreting the Latent Space of GANs for Semantic Face Editing". In: *CVPR*.

Shen, Yujun and Zhou, Bolei (Dec. 2020). "Closed-Form Factorization of Latent Semantics in GANs". In: *arXiv:2007.06600 [cs]*. arXiv: 2007.06600. URL: `http://arxiv.org/abs/2007.06600` (visited on 12/18/2020).

Shoemake, Ken (July 1985). "Animating rotation with quaternion curves". In: *ACM SIGGRAPH Computer Graphics* 19.3, pp. 245–254. ISSN: 0097-8930. DOI: `10.1145/325165.325242`. URL: `https://doi.org/10.1145/325165.325242` (visited on 12/06/2022).

Simonyan, Karen and Zisserman, Andrew (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*.

Skorokhodov, Ivan, Tulyakov, Sergey, and Elhoseiny, Mohamed (Dec. 2021). "StyleGAN-V: A Continuous Video Generator with the Price, Image Quality and Perks of StyleGAN2". In: *arXiv:2112.14683 [cs]*. arXiv: 2112.14683. URL: `http://arxiv.org/abs/2112.14683` (visited on 01/04/2022).

– (Apr. 2022). "StyleGAN-V: A Continuous Video Generator with the Price, Image Quality and Perks of StyleGAN2". In: *arXiv:2112.14683 [cs]*. arXiv: 2112.14683. URL: `http://arxiv.org/abs/2112.14683` (visited on 04/28/2022).

Sohl-Dickstein, Jascha, Weiss, Eric, Maheswaranathan, Niru, and Ganguli, Surya (July 2015). "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 2256–2265.

Song, Linsen, Wu, Wayne, Qian, Chen, He, Ran, and Loy, Chen Change (2020). "Everybody's Talkin': Let Me Talk as You Want". In: *arXiv preprint* arXiv:

Song, Yang, Zhu, Jingwen, Li, Dawei, Wang, Xiaolong, and Qi, Hairong (July 2019). *Talking Face Generation by Conditional Recurrent Adversarial Network*. arXiv:1804.04786 [cs]. DOI: `10.48550/arXiv.1804.04786`. URL: `http://arxiv.org/abs/1804.04786` (visited on 03/29/2023).

Stypułkowski, Michał, Vougioukas, Konstantinos, He, Sen, Zieba, Maciej, Petridis, Stavros, and Pantic, Maja (July 2023). *Diffused Heads: Diffusion Models Beat GANs on Talking-Face Generation*. arXiv:2301.03396 [cs]. DOI: 10.48550/arXiv.2301.03396. URL: http://arxiv.org/abs/2301.03396 (visited on 08/03/2023).

Suwajanakorn, Supasorn, Seitz, Steven M., and Kemelmacher-Shlizerman, Ira (July 2017a). "Synthesizing Obama: Learning Lip Sync from Audio". In: *ACM Trans. on Graph. (Proceedings of SIGGRAPH)* 36.4.

– (July 2017b). "Synthesizing Obama: learning lip sync from audio". In: *ACM Transactions on Graphics* 36.4, 95:1–95:13. ISSN: 0730-0301. DOI: 10.1145/3072959.3073640. URL: https://doi.org/10.1145/3072959.3073640 (visited on 05/11/2022).

Taylor, Sarah, Kim, Taehwan, Yue, Yisong, Mahler, Moshe, Krahe, James, Rodriguez, Anastasio Garcia, Hodgins, Jessica, and Matthews, Iain (Aug. 2017). "A deep learning approach for generalized speech animation". en. In: *ACM Transactions on Graphics* 36.4, pp. 1–11. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3072959.3073699. URL: https://dl.acm.org/doi/10.1145/3072959.3073699 (visited on 08/05/2023).

Thies, Justus, Elgharib, Mohamed, Tewari, Ayush, Theobalt, Christian, and Nießner, Matthias (July 2020a). "Neural Voice Puppetry: Audio-driven Facial Reenactment". In: *arXiv:1912.05566 [cs]*. arXiv: 1912.05566. URL: http://arxiv.org/abs/1912.05566 (visited on 11/11/2021).

– (2020b). "Neural Voice Puppetry: Audio-driven Facial Reenactment". In: *ECCV 2020*.

Tian, Yu, Ren, Jian, Chai, Menglei, Olszewski, Kyle, Peng, Xi, Metaxas, Dimitris N., and Tulyakov, Sergey (Apr. 2021). "A Good Image Generator Is What You Need for High-Resolution Video Synthesis". In: *arXiv:2104.15069 [cs]*. arXiv: 2104.15069. URL: http://arxiv.org/abs/2104.15069 (visited on 12/10/2021).

Tulyakov, Sergey, Liu, Ming-Yu, Yang, Xiaodong, and Kautz, Jan (Dec. 2017). "MoCoGAN: Decomposing Motion and Content for Video Generation". In: *arXiv:1707.04993 [cs]*. arXiv: 1707.04993. URL: http://arxiv.org/abs/1707.04993 (visited on 12/10/2021).

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz, and Polosukhin, Illia (Dec. 2017). "Attention Is All You Need". In:

arXiv:1706.03762 [cs]. arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (visited on 08/02/2020).

Vondrick, Carl, Pirsiavash, Hamed, and Torralba, Antonio (Oct. 2016). *Generating Videos with Scene Dynamics*. arXiv:1609.02612 [cs]. DOI: 10.48550/arXiv.1609.02612. URL: http://arxiv.org/abs/1609.02612 (visited on 05/10/2023).

Vougioukas, Konstantinos, Petridis, Stavros, and Pantic, Maja (2018a). "End-to-End Speech-Driven Facial Animation with Temporal GANs". In: *BMVC*.

– (July 2018b). "End-to-End Speech-Driven Facial Animation with Temporal GANs". In: *arXiv:1805.09313 [cs, eess]*. arXiv: 1805.09313. URL: http://arxiv.org/abs/1805.09313 (visited on 06/22/2021).

– (Oct. 2019). "Realistic Speech-Driven Facial Animation with GANs". In: *International Journal of Computer Vision (IJCV)*.

Voynov, Andrey and Babenko, Artem (June 2020). *Unsupervised Discovery of Interpretable Directions in the GAN Latent Space*. arXiv:2002.03754 [cs, stat]. DOI: 10.48550/arXiv.2002.03754. URL: http://arxiv.org/abs/2002.03754 (visited on 07/16/2023).

Wang, Binxu and Ponce, Carlos R. (Mar. 2021). *The Geometry of Deep Generative Image Models and its Applications*. arXiv:2101.06006 [cs, math]. DOI: 10.48550/arXiv.2101.06006. URL: http://arxiv.org/abs/2101.06006 (visited on 07/16/2023).

Wang, Jiadong, Qian, Xinyuan, Zhang, Malu, Tan, Robby T., and Li, Haizhou (June 2023). "Seeing What You Said: Talking Face Generation Guided by a Lip Reading Expert". en. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada: IEEE, pp. 14653–14662. ISBN: 9798350301298. DOI: 10.1109/CVPR52729.2023.01408. URL: https://ieeexplore.ieee.org/document/10204342/ (visited on 12/12/2023).

Wang, Kaisiyuan, Wu, Qianyi, Song, Linsen, Yang, Zhuoqian, Wu, Wayne, Qian, Chen, He, Ran, Qiao, Yu, and Loy, Chen Change (2020). "Mead: A large-scale audio-visual dataset for emotional talking-face generation". In: *European Conference on Computer Vision*. Springer, pp. 700–717.

Wang, Suzhen, Li, Lincheng, Ding, Yu, and Yu, Xin (Dec. 2021). *One-shot Talking Face Generation from Single-speaker Audio-Visual Correlation Learning*. en. arXiv:2112.02749 [cs]. URL: http://arxiv.org/abs/2112.02749 (visited on 08/05/2023).

Wang, Zhou, Bovik, A.C., Sheikh, H.R., and Simoncelli, E.P. (Apr. 2004). "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4. Conference Name: IEEE Transactions on Image Processing, pp. 600–612. ISSN: 1941-0042. DOI: 10.1109/TIP.2003.819861.

Wei, Tianyi, Chen, Dongdong, Zhou, Wenbo, Liao, Jing, Zhang, Weiming, Yuan, Lu, Hua, Gang, and Yu, Nenghai (2022). "E2Style: Improve the Efficiency and Effectiveness of StyleGAN Inversion". In: *IEEE Transactions on Image Processing* 31. arXiv:2104.07661 [cs], pp. 3267–3280. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2022.3167305. URL: http://arxiv.org/abs/2104.07661 (visited on 12/14/2023).

White, Tom (Dec. 2016). *Sampling Generative Networks*. arXiv:1609.04468 [cs, stat]. DOI: 10.48550/arXiv.1609.04468. URL: http://arxiv.org/abs/1609.04468 (visited on 11/21/2023).

Xia, Weihao, Zhang, Yulun, Yang, Yujiu, Xue, Jing-Hao, Zhou, Bolei, and Yang, Ming-Hsuan (Aug. 2021). "GAN Inversion: A Survey". In: *arXiv:2101.05278 [cs]*. arXiv: 2101.05278. URL: http://arxiv.org/abs/2101.05278 (visited on 01/20/2022).

– (Mar. 2022). "GAN Inversion: A Survey". In: *arXiv:2101.05278 [cs]*. arXiv: 2101.05278. URL: http://arxiv.org/abs/2101.05278 (visited on 05/10/2022).

Xie, Lei and Liu, Zhi-Qiang (Aug. 2007a). "A coupled HMM approach to video-realistic speech animation". In: *Pattern Recognition* 40, pp. 2325–2340. DOI: 10.1016/j.patcog.2006.12.001.

– (May 2007b). "Realistic Mouth-Synching for Speech-Driven Talking Face Using Articulatory Modelling". In: *Multimedia, IEEE Transactions on* 9, pp. 500–510. DOI: 10.1109/TMM.2006.888009.

Xu, Yangyang, Du, Yong, Xiao, Wenpeng, Xu, Xuemiao, and He, Shengfeng (Aug. 2021). "From Continuity to Editability: Inverting GANs with Consecutive Images". In: *arXiv:2107.13812 [cs]*. arXiv: 2107.13812. URL: `http://arxiv.org/abs/2107.13812` (visited on 05/10/2022).

Yamamoto, Eli, Nakamura, Satoshi, and Shikano, Kiyohiro (1998). "Lip movement synthesis from speech based on Hidden Markov Models". In: *Speech Communication* 26.1-2, pp. 105–115.

Yi, Ran, Ye, Zipeng, Zhang, Juyong, Bao, Hujun, and Liu, Yong-Jin (Mar. 2020). "Audio-driven Talking Face Video Generation with Learning-based Personalized Head Pose". In: *arXiv:2002.10137 [cs]*. arXiv: 2002.10137 version: 2. URL: `http://arxiv.org/abs/2002.10137` (visited on 05/30/2020).

Yu, Zhentao, Yin, Zixin, Zhou, Deyu, Wang, Duomin, Wong, Finn, and Wang, Baoyuan (Dec. 2022). *Talking Head Generation with Probabilistic Audio-to-Visual Diffusion Priors*. arXiv:2212.04248 [cs, eess]. DOI: `10.48550/arXiv.2212.04248`. URL: `http://arxiv.org/abs/2212.04248` (visited on 12/13/2023).

Yushchenko, Vladyslav, Araslanov, Nikita, and Roth, Stefan (Oct. 2019). "Markov Decision Process for Video Generation". en. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Seoul, Korea (South): IEEE, pp. 1523–1532. ISBN: 978-1-72815-023-9. DOI: `10.1109/ICCVW.2019.00190`. URL: `https://ieeexplore.ieee.org/document/9022386/` (visited on 05/10/2023).

Zhang, Richard, Isola, Phillip, Efros, Alexei A., Shechtman, Eli, and Wang, Oliver (Apr. 2018). "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *arXiv:1801.03924 [cs]*. arXiv: 1801.03924. URL: `http://arxiv.org/abs/1801.03924` (visited on 12/16/2020).

Zhang, Zhimeng, Li, Lincheng, Ding, Yu, and Fan, Changjie (2021). "Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3661–3670.

Zhong, Weizhi, Fang, Chaowei, Cai, Yinqi, Wei, Pengxu, Zhao, Gangming, Lin, Liang, and Li, Guanbin (June 2023). "Identity-Preserving Talking Face Generation with Landmark and Appearance Priors". en. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern*

Recognition (CVPR). Vancouver, BC, Canada: IEEE, pp. 9729–9738. ISBN: 9798350301298. DOI: 10.1109/CVPR52729.2023.00938. URL: https://ieeexplore.ieee.org/document/10203630/ (visited on 12/12/2023).

Zhou, Hang, Liu, Yu, Liu, Ziwei, Luo, Ping, and Wang, Xiaogang (Apr. 2019). "Talking Face Generation by Adversarially Disentangled Audio-Visual Representation". In: *arXiv:1807.07860 [cs]*. arXiv: 1807.07860. URL: http://arxiv.org/abs/1807.07860 (visited on 05/24/2021).

Zhou, Yang, Han, Xintong, Shechtman, Eli, Echevarria, Jose, Kalogerakis, Evangelos, and Li, Dingzeyu (Oct. 2020a). "MakeItTalk: Speaker-Aware Talking-Head Animation". In: *arXiv:2004.12992 [cs]*. arXiv: 2004.12992. URL: http://arxiv.org/abs/2004.12992 (visited on 12/16/2020).

– (2020b). "MakeItTalk: Speaker-Aware Talking-Head Animation". In: *ACM Transactions on Graphics* 39.6.

Zhu, Jun-Yan, Krähenbühl, Philipp, Shechtman, Eli, and Efros, Alexei A. (Dec. 2018). *Generative Visual Manipulation on the Natural Image Manifold*. arXiv:1609.03552 [cs] version: 2. DOI: 10.48550/arXiv.1609.03552. URL: http://arxiv.org/abs/1609.03552 (visited on 11/04/2022).

Zhu, Peihao, Abdal, Rameen, Qin, Yipeng, Femiani, John, and Wonka, Peter (Oct. 2021). *Improved StyleGAN Embedding: Where are the Good Latents?* arXiv:2012.09036 [cs] version: 3. DOI: 10.48550/arXiv.2012.09036. URL: http://arxiv.org/abs/2012.09036 (visited on 12/14/2023).