Controllable GANs applied to 3D face generation - Literature review

Liam Watson
Supervisor: Dr. D. Moodley

October 16, 2022

Contents

1 Abstract **2**

2 Introduction **2**

3 Intuition behind GANs **2**
3.1 Nash Equilibrium . . . 2
3.2 Latent space . . . 2

4 Problem statement and applications **2**

5 3D model representations **3**
5.1 Point clouds . . . 3
5.2 Triangle mesh . . . 3
5.3 Voxels . . . 3

6 Variety of GANs **3**
6.1 Convolutional . . . 3
6.2 Wasserstein GAN (WGAN) . . . 3
6.3 Progressive GANs . . . 3
6.4 Conditional GANs . . . 4
6.5 Controllable GANs . . . 4

7 Evaluation metrics **4**
7.1 Quantitative . . . 4
7.1.1 Fréchet Inception Distance (FID)[19] . . . 4
7.1.2 KID . . . 4
7.1.3 Generalisation . . . 5
7.1.4 Specificity . . . 5
7.2 Qualitative . . . 5
7.2.1 Nearest Neighbors . . . 5
7.2.2 Rapid Scene Categorization . . . 5
7.2.3 Preference Judgement . . . 5
7.2.4 Mode Drop and Collapse . . . 5

8 Work done on controllable GANs 3D Faces **5**
8.0.1 InfoGAN . . . 5
8.0.2 HoloGAN . . . 6
8.1 Mesh GAN . . . 7
8.1.1 3DMMs . . . 7
8.1.2 MeshGAN concepts and Method . . . 7

9 Problems with GANs **8**
9.1 Mode collapse . . . 8
9.2 Vanishing gradients . . . 8
9.3 Convergence failure . . . 8
9.4 Generation output quality . . . 9

10 Datasets **9**

11 Conclusion **9**

1 Abstract

This paper describes the current state of 3D face synthesis using controllable GANs. Following the intuition behind the minimax game proposed in a GAN, the underpinning concepts required for controllable GANs are shown with an explanation of the problem domain as well as the applications of 3D face generation and manipulation. Further we discuss the different representations of 3D faces, including why one representation would be advantageous over another. We describe the variety of GANs that have been used for 2D and 3D face generation as well as delve deep into the normative structure of controllable GANs preceding a review and discussion of the state of the art techniques for generating faces with focus on a models understanding of the 3D world. The many problems with generating 3D faces that have been studied in the literature are explained along with proposed solutions. Finally the current state of 3D facial data is covered with explanation of some of the flaws.

2 Introduction

Modern machine learning techniques have been able to solve many problems that cannot be easily solved by hand crafted techniques, however, generation had remained a stubborn problem until the advent of Generative Adversarial Networks (GANs) [17]. There has been remarkable progress using GANs on 2D images for arbitrary face generation and feature manipulation [25]. However, the three dimensional domain has remained relatively untouched. The majority of techniques working on three dimensional face generation have focused on developing models with an understanding of the 3D world but producing 2D output with controllable pose.[22]. Recently methods have been proposed with promising results working purely in the three dimensional domain such as MeshGAN [10] with the large majority of research being focused on a hybrid learning architecture combining 3D Morphable Models (3DMMs) in conjunction with another deep learning technique [37].

3 Intuition behind GANs

GANs were developed by Ian Good Fellow in 2014, they noticed that discriminative models had made rapid progress with deep generative models having little progress due difficulty approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and difficulty of leveraging the benefits of piecewise linear units in the generative context.[17] The proposed solution was to leverage natural competition by pitting a generative model against a discriminative, which negates the previous difficulties generative research had encountered. Both models attempt to beat the other with the generator consistently creating data that is more likely to fool the discriminator and the discriminator improves its ability to determine if the generated data is synthetic.
More rigorously the competition can be described as the generator attempting to form probability distribution $P_G(x)$ that mimics the true data distribution $P_{data}(x)$. The generator attempts to form the distribution from random noise information $z \sim P_{noise}(x)$. The generator is trained against the discriminator that aims to separate real distribution P_{data} and generated distribution P_G. The optimal discriminator can be described as $D(x) = \frac{P_{data}(x)}{P_{data}(x)+P_G(x)}$. With the entire game being expressed mathematically as:

$$\min_G \max_D V(D,G)$$
$$= \mathbb{E}_{x\sim P_{data}} [\log D(x)] + \mathbb{E}_{z\sim P_{noise}} [\log (1 - D(G(z)))]$$

3.1 Nash Equilibrium

The minimax game that the two models engage in hopes to settle on an optimal set of model parameters for the given problem, a so called Nash Equilibrium. It has been shown in the literature that for simple GANs as well as more complex Wasserstein GANs (WGANs [2]) that models do not converge to an exact Nash Equilibrium [15], however, it is common for machine learning techniques such as MLPs or CNNs to converge to a local optimality.

3.2 Latent space

An important concept in GANs is latent space which is an embedding data within a feature space where items which closely resemble each other have a shorter distance between each other. In GANs we treat the latent space as a hypothetical region of data that is interpreted and explored by the generator. This interpretation of the input can be leveraged in order to isolate and disentangle features or effectively generate similar output by intelligently moving within the latent domain[31].

4 Problem statement and applications

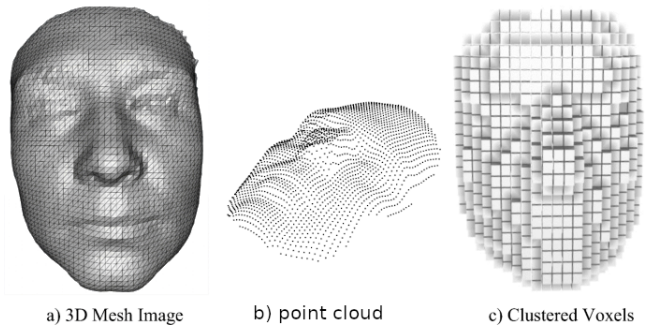
GANs have been extensively used for 2D face generation with astonishing results in papers like styleGAN [25], however, modern computational power has enabled highly detailed environments in three dimensions. Consumers in the Anthropocene era demand highly realistic 3D models for video games, movies and commercial engagement. In order to overcome the uncanny valley, a method to produce high fidelity, feature controlled human faces is needed. Such a technique will enable many innovations in commercial applications such as virtual reality avatars, augmented reality, video game development, teleconferencing, virtual try-on, special effects in movies, with potentially many more applications as faces are ubiquitous.
In particular controllable GANs show big promise in the literature for this task - generating arbitrary three dimensional feature controlled human faces.

The first use of GANs was in natural image generation such as hand written digits and photographs [17]. However the field is now large and varied with applications in medical image analysis [35], image editing [41], 3D model generation, asset generation for digital entertainment [23][36], image semantic segmentation [18] and more [8]. The reason GANs have so many applications is because generation is a problem that is difficult to solve with traditional hand grafted techniques and GANs handle abstract continuous data well because the discriminator acts as a continually improving evaluation metric.

5 3D model representations

In order to work with 3D faces we first need to understand how 3D data is represented and stored. There are three main representations used in the literature, namely point clouds, voxels and triangular meshes.

Figure 1: Point cloud 3D face model [14, 38]



5.1 Point clouds

Point clouds are a simple approach to 3D model representation consisting of many three dimensional infinitesimal points in space forming an object. This strategy is intuitive, accurate with enough points and relies on basic mathematical operations, however, can be very memory intensive due to the large number of points required [14].

5.2 Triangle mesh

Expanding on the point cloud strategy meshes combine points in space into simple polygons - primarily triangles - which define a plane in three dimensional space. This approach can be used to more accurately represent models with fewer points. The number of points required for an accurate representation is reduced massively for some shapes but curves must be approximated. Memory requirements for detailed models are still high and more computational expense is needed for operations on the model due to processing individual polygons [39].

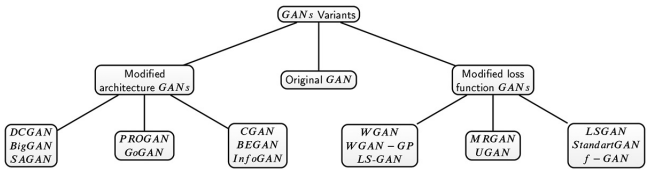
5.3 Voxels

Voxels are a newer technique that leave points in three dimensional space behind and choose to represent models with, what for all intensive purposes is a pixel extended to the third dimension. Memory requirements are typically lower however there are some known issues with detail representation [38].

6 Variety of GANs

Since the inception of GANs there have been many proposed additions and modifications to the architecture and loss function which address the many problems with the original GAN while attempting to improve generated results.

Figure 2: Taxonomy of GANs [40]



6.1 Convolutional

The original GAN uses MLP networks which require us to flatten data for it to be fed into a network, inherently disregarding positional information that may be stored in the arrangement of the data elements. A CNN takes a natural approach to resolve this issue which was inspired by biology - rather than processing linear data, the network accepts tensor data. CNNs have been shown to out-perform MLPs in many applications due to the preservation of relative data element distance - aggregating many elementary features into a more complex understanding of data[27]. A GAN that implements convolutional layers is called a deep convolutional GAN (DCGAN) and is generally considered to out perform Standard GANs especially in image applications [32].

6.2 Wasserstein GAN (WGAN)

The original GAN loss function is inherintly flawed in that the game loss function is somewhat ill phrased giving rise to training problems such as mode collapse and vanishing gradients. The WGAN paper proposes moving away from the standard minimax game loss fuction to the well defined Wasserstein-1 distance, the resulting loss function is described by

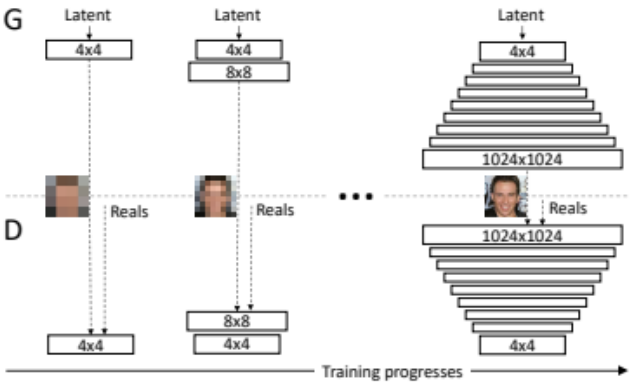
$$L_{WGAN} = inf_{\gamma \in \Pi(P_r, P_g)} \mathbf{E}_{(x_r, x_g) \sim \gamma}[x_r - x_g]$$

The new loss function dramatically improves training stability and can now be used as a termination criterion but has been shown to cause poor sample generation and convergence problems. Additionally it should be noted that this metric is more expensive to compute and can increase training time. [22]

6.3 Progressive GANs

Progressive GANs seek to increase the generated sample resolution which is typically low in a simple GAN. The novel training technique used in ProGANs is that the generator and discriminator progressively grow from a low level of detail to a high level as training evolves. Both model training speed and stability are greatly improved yielding highly detailed generated samples. [24]

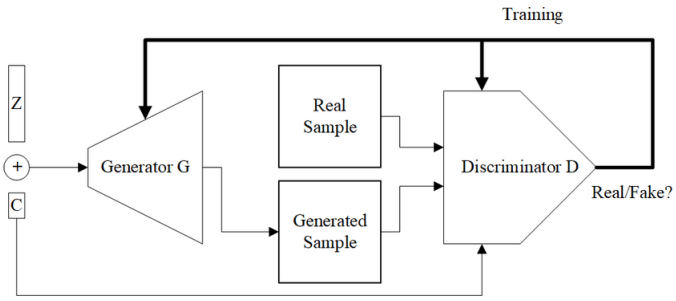
Figure 3: Architecture of ProGAN [24]



6.4 Conditional GANs

Conditional GANs were first proposed as a method to enable feature control over generated samples. The archutecture in addition to the latent vector requires an additional label code in order to manipulate features. The technique requires labelled data in the training dataset in order to learn the desired feature classes which presents a large dataset restriction. There have been issues with this architecture failing to disambiguate features especially in more complex problems such as facial expressions.[22]

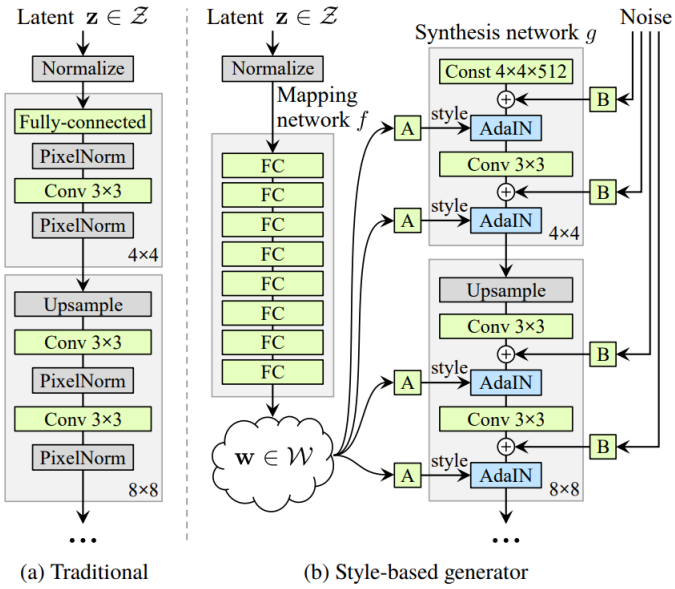
Figure 4: Architecture of conditional GAN [30]



6.5 Controllable GANs

The main drawback of conditional GANs is that they require detailed labeling of training data, controllable GANs seek to circumvent this restraint while still retaining feature control. The strategy for feature control is to tweak the input latent vector to intelligently locate the desired featureset within the latent space. Typically a pretrained classifier is used to check for the presence of desired features within the synthetic samples in order to refine changes to the latent vector.

Figure 5: Normative architecture of controllable GAN [26]



The architecture above has been the most well documented controllable GAN in the literature deemed StyleGAN and has had the source code released on GitHub.

7 Evaluation metrics

There are numerous evaluation techniques used for GANs most of which can be found in "Pros and cons of gan evaluation measures"[6]. In this paper we will cover the most commonly used in recent research on 3D face generation.

7.1 Quantitative

7.1.1 Fréchet Inception Distance (FID)[19]

FID was first introduced by Heusel, Ramsauer, Unterthiner, Nessler, and Hochreiter in 2017. The technique embeds a set of generated samples into a feature space given by a specific layer of the Inception network (a model that embeds data into a feature space). Viewing the layer as a continuous multivariate Gaussian, the mean and covariance are estimated for both the real and generated data. The Fréchet distance between these two gaussians (Wasserstein-2 distance) is then used to quantify the quality of generated samples. One can express this evaluation as the following formula:

$$FID(r,g) = ||\mu_r - \mu_g||_2^2 + Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}\right) \quad (1)$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the mean and covariance of the real data and generated sample distributions respectively. Lower FID is better as there is a smaller distance between the generated and real data distributions. It has been shown that FID performs well in discriminability, robustness, computational efficiency and has been shown to agree with human judgements[19].

7.1.2 KID

Kernel Inception Distance (KID) is an unbiast metric unlike FID and rather computes the squared maximum mean discrepancy between feature representations (computed from the Inception model as described for FID) of the real and generated images. [4, 29]

7.1.3 Generalisation

Generalisation measures the models ability to represent unseen face shapes that it did not encounter during training. Computation of generalisation error is done by per-vertex Euclidean distance between samples in the test set $\mathbf{x}$ and the generators reconstruction $\mathbf{x}^*$ which is expressed $\mathbf{x}^* = \text{argmin}_z |\mathbf{x} - G(\mathbf{z})|$. The overall metric for the model is found by averaging over all the vertices and test samples. [7]

7.1.4 Specificity

Specificity evaluates the validity of faces generated by the model. An arbitrarily large number of generated faces are synthesised and proximity to real faces in the test dataset is measured. The proximity is calculated using the Euclidean distance described in generalisation. This metric can provide evidence that faces generated by a model are comparatively more realistic than other results. [5]

7.2 Qualitative

There are typically less qualitative evaluation metrics than quantitative, however, such metrics are important as generated faces must be of high quality when reviewed by humans in order to overcome human sensitivity to faces - the so called uncanny valley. Qualitative metrics can additionally be important for detecting training performance and failure such as over fitting or mode collapse.

7.2.1 Nearest Neighbors

Nearest neighbors analysis has been used in the literature to detect overfitting in a training set by showing synthetic samples to their nearest neighbors in the training set. Typically nearest neighbors is determined using Euclidean distance which is sensitive to small perturbations in perception. Models that store training images can circumvent this training trivially but this can be elevated by choosing nearest neighbors based on perceptual measures and selecting many nearest neighbors.

7.2.2 Rapid Scene Categorization

RSC is used to obtain a measure of synthetic models quality by rapidly showing participants a mixture of real and fake images, they are asked to distinguish the two sample types. This technique is good as it intuitively seeks to answer the penultimate question - is the generated sample of human standard. However this method has issues around experimental conditions and bias due to its physical human nature.

7.2.3 Preference Judgement

Preference judgement asks participants to evaluate generated images in terms of quality. Much like RSC this metric attempts to determine if results are acceptable to a human audience but suffer from similar experimental design problems.

7.2.4 Mode Drop and Collapse

There have been many proposed quantitative measure for mode collapse, however, in certain small data size circumstances a nearest neighbor-like investigation can be useful to truly detect mode collapse.

8 Work done on controllable GANs 3D Faces

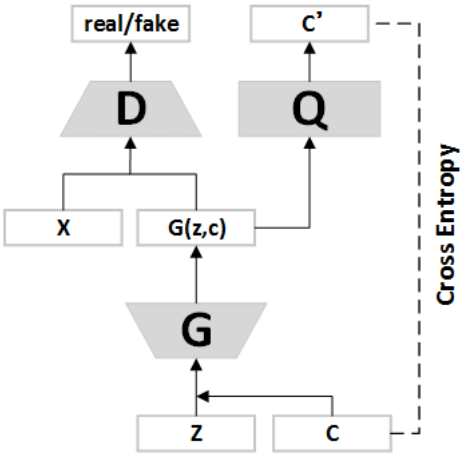
There have been three well recognised papers on 3D face generation and manipulation using controllable GANs - namely Info-GAN [9], Holo-gan [29] and MeshGAN[10].

8.0.1 InfoGAN

InfoGAN was the first controllable GAN technique to attempt 3D face generation and manipulation. A simple GAN uses a factored continuous input noise vector and does not impose restrictions on how the generator may use this noise. Due to this the generator may tend to entangle dimensions of the noise vector z resulting in poor recognition of semantic data features. InfoGAN decomposes the simple noise vector into a source of incompressible noise z and the latent code c which aims to parameterise semantic features of the data distribution. Rigorously one can denote the set of structured latent variables $C = \{c_1, c_2, ..., c_L\}$. The paper assumes a factored distributions $P(c_1, c_2, ..., c_L) = \Pi_{i=0}^{L} P(c_i)$. The generator is then a function of the two latent variables z and c. The paper suggests using information theory in order to rigorously define a controllable GAN. It is proposed that a standard GAN architecture would find an optimal solution by ignoring the latent code which satisfies $P_G(x|c) = P_G(x)$. The solution to this trivial convergence is to use information-theoretic regularization - mutual information between the generators distribution and the latent codes should be high. $I(c; G(z, c))$ should be high. The new game cost function proposed by the paper is

$$\min_G \max_D V_I(D, G) = V(D, G) - \lambda I(c; G(z, c))$$

Figure 6: The architecture of InfoGAN showing [28]

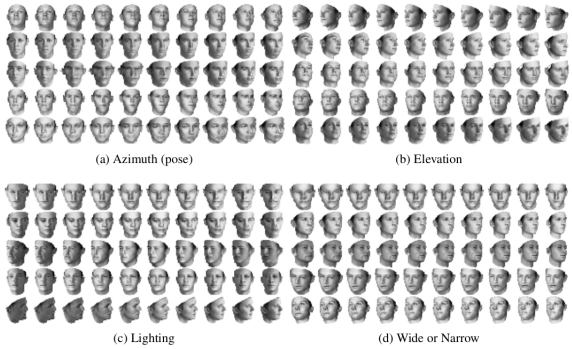


The paper notes that the mutual information between c and $G(z, c)$ is difficult to maximise in practice so it is proposed that a lower bound is defined. Using Variational Information Maximization [1] and and a simplifying lemma the paper finds InfoGANs game function to be

$$\min_{G,Q} \max_D V_{InfoGAN}(D, G, Q) = V(D, G) - \lambda L_I(G, Q)$$

Where $Q(c|x)$ is an approximation of $P(c|x)$, $L_I(G,Q)$ the variational lower bound that is maximally calculated using a Monte Carlo simulation and lambda is hyperparameter that controls the influence of L_I.
During implementation the team parameterises Q as a neural network sharing all convolutional layers with the discriminator but with a fully connected layer to convert output parameters. The latent codes c_i of categorical form are generated using softmax nonlinearity for Q but there are several choices for the continuous latent codes generated for P. The team finds setting $\lambda = 1$ a sufficient condition for convergence.

Figure 7: InfoGAN results varying continuous latent factors from -1 to 1. [9]



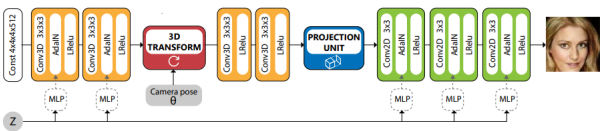
InfoGAN produced reasonable quality results for a fully unsupervised technique which works on unlabelled data and self discovers features while disentangling them. Additionally InfoGAN comes at negligible computational complexity over a typical GAN architecture. However, the results are of lower resolution and have unwanted aberrations which are particularly noticeable in the results figure 7. Further the method does not directly address 3D faces nor does the model have any explicit understanding of the three dimensional world. The source code for InfoGAN was released by the team and can be found on GitHub which will make further work using their information theory techniques easier.

8.0.2 HoloGAN

HoloGAN introduces an advancement over InfoGAN for unsupervised learning of 3D representations from natural images. The new approach recognises that prior techniques use 2D kernels to generate and make assumptions about a 3D environment which causes generated samples to be blurry and contain artefacts. Rather than using 2D kernels HoloGAN learns a true 3D representation of the world which can then be manipulated via transformations and brought down into 2D through a projection.

The advancements proposed by HoloGAN are control over which semantic the model will learn, better quality output and guaranteed semantic value for latent codes.

Figure 8: HoloGAN architecture [29]



The architecture and technique of HoloGAN can be constructed from a model learning 3D representations, a Projection unit, a model learning with view-dependent mappings and new loss functions.

The primary input to HoloGAN is a constant learnt 4×4×4×512 tensor with the noise vector $\mathbf{z}$ being used to inject style into the model. The style is injected - mapped to affine parameters for adaptive instance normalization (AdaIN) [20]- after each convolutional layer using an MLP. AdaIN is defined as follows:

$$\text{AdaIN}(\Phi_l(\mathbf{x},\mathbf{z}) = \sigma(z)\left(\frac{\Phi_l(\mathbf{x}) - \mu(\Phi_l(\mathbf{x}))}{\sigma(\Phi_l(\mathbf{x}))}\right) + \gamma(\digamma)$$

With Φ_l being some features at a layer l of an image $\mathbf{x}$. AdaIN is used to conform to the mean and standard deviation of features at the different levels l. This strategy was first proposed in the paper StyleGAN in 2D rather than 3D. [25]

More bias about the 3D environment is added to HoloGAN during training by explicitly transforming the learnt 3D features to some random pose before projection into 2D. This strategy ensures that the model learns a 3D representation that is disentangled from other features and can be rendered from all view points.

HoloGAN trains a differentiable projection unit in order to extract meaningful 3D representations from 2D images despite occlusions that need information to be filled into a 3D model. The projection unit processes the 3D features that have been pre-processed from the learnt tensor and outputs 2D features.
HoloGAN takes some novel approaches to loss functions in addition to high level architecture. The model uses an identity regulariser $l_{identity}$ that ensures generated vectors match the latent vector $\mathbf{z}$. This strategy allows higher resolution generation and encourages the model to use $\mathbf{z}$ to maintain identity through pose changes.

$$L_{identity}(G) = \mathbb{E}_{\mathbf{z}}||\mathbf{z} - F(G(\mathbf{z}))||^2$$

Where F is introduced as an encoder network similar to the discriminator that uses a fully connected layer to predict the reconstructed latent vector.

Holo-GAN additionally introduces multi-scale style discriminators that perform the same function as the generator which learns style of training images but rather at the feature level. More rigorously the style discriminator classifies the mean $\mu(\Phi_l)$ and standard deviation $\sigma(\Phi_l)$ at each level which prevents mode collapse thus enabling longer training. The style discriminator D_l and loss L^l_{style} at layer l can be defined as:

$$D_l(\mathbf{x}) = \tilde{D}_l(\mu(\Phi_l(\mathbf{x})), \sigma(\Phi_l((x))))$$
$$L^l_{style}(G) = \mathbb{E}_{\mathbf{x}}\left[-\log D_l(G(\mathbf{z}))\right]$$

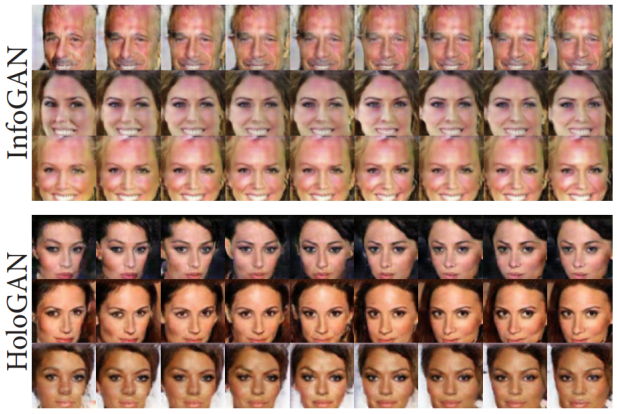
Then the total loss for HoloGAN can be expressed as:

$$L_{total}(G) = L_{GAN}(G) + \lambda_i \cdot L_{identity}(G) + \lambda_s \cdot \sum_l L^l_{style}(G)$$

Where L_{GAN} is the GAN loss used in DC-GAN.[32]

The results from HoloGAN are high quality with little to no aberrations.

Figure 9: HoloGAN results on generating faces [29]



Method	CelebA	Chairs	Cars
DCGAN	1.81 ± 0.09	6.36 ± 0.16	4.78 ± 0.11
LSGAN	1.77 ± 0.06	6.72 ± 0.19	4.99 ± 0.13
WGAN-GP	**1.63 ± 0.09**	9.43 ± 0.24	15.57 ± 0.29
HOLOGAN	2.87 ± 0.09	**1.54 ± 0.07**	**2.16 ± 0.09**

Table 1: Showing KID between real images and images generated by Holo-GAN and other 2D GANs (lower is better)

We can see from Figure 9 that HoloGan is able to produce high fidelity realistic faces that can be manipulated in three dimensional space. When compared to InfoGAN results one can see that synthetic faces are more appealing with fewer to no detectable visual distortions or aberrations. Quantitatively the team compares HoloGAN to three other techniques in table 1 where we can see that HoloGAN performs poorly on celebrity faces but favorably on chairs and cars. This result is still impressive as HoloGAN can provide visually appealing results while still maintaining control over a given set of features. The team has released the source code for HoloGAN which makes future research on improvements and verification much easier - it can be found on GitHub

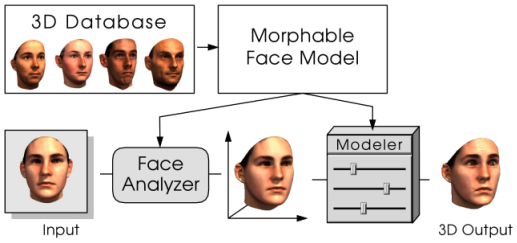
8.1 Mesh GAN

All the techniques covered up until now have been focused fundamentally on generating a 2D output sample even though the model learns a representation of the 3D world these models are flawed as they do not give us the freedom a 3D model does. MeshGAN implements several new strategies for training on 3D model input and producing a fully three dimensional output - it is the first intrinsically 3D GAN architecture although not in the normal form of a controllable GAN but does maintain feature control.
MeshGAN proposes the use of 3D Morphable Models (3DMMs) using Principal Component Analysis where a model is used as an input to the network and the network proposes manipulations to change to model.

8.1.1 3DMMs

3DMMs are fundamentally a method that presents a representation of a 3D face but, with some added statistical complexity that is used to simplify the concept of a face. The key principle is based on two key ideas: Faces typically have a high point correspondence which can be maintained. This correspondence can be leveraged in order to generate novel faces using linear combinations of other faces producing realistic faces - so called morphs. Second, the underlying facial structure is separate from tertiary factors such as lighting. MeshGAN uses a true triangular mesh data representation which is defined in the paper as follows. Morphable models combine these ideas with PCA in order to form an understanding of a face which can be tweaked by a model such as MeshGAN.

Figure 10: 3DMMs used in deep learning facial generation [29]



8.1.2 MeshGAN concepts and Method

Facial surface is a manifold triangular mesh $\mathscr{M} = (\{1,..,n\}, \varepsilon = \varepsilon_i \cup \varepsilon_b, \mathscr{F})$ where each edge $e_{ij} \in \varepsilon$ belongs to at most two triangular faces F_{ijk} and F_{jih}. The embedding of $\mathscr{M}$ is created using an $n \times 3$ matrix $\mathbf{V}$ containing vertex coordinates on the rows. The discrete Riemannian metric is defined by assigning length $l_{ij} = ||\mathbf{v}_i - \mathbf{v}_j||^2$ to each edge. Further the paper discretises the Laplacian as $\nabla = \mathbf{A}^{-1}\mathbf{W}$ using the distance equivalent of the cotangent formula described in "A cotangent laplacian for images as surfaces" [21].

The paper then describes spectral mesh convolutions with $\mathbf{f} = (f_1, ..., f_n)^T$ being a scalar function on the vertices of the mesh. This function admits a Fourier decomposition $\mathbf{f} = \Phi\Phi^T\mathbf{A}\mathbf{f}$ which can be used to describe the convolution operation in the spectral domain as $\mathbf{f} \circledast \mathbf{g} = \Phi(\hat{\mathbf{f}} \cdot \hat{\mathbf{f}}) = \Phi(\Phi^T\mathbf{A}\mathbf{f}) \cdot (\Phi^T\mathbf{A}\mathbf{g})$.

This Fourier decomposition is then used to define a graph convolutional neural network where the spectral convolution operation is of the form $\mathbf{f}' = \Phi\hat{\mathbf{G}}\Phi^T\mathbf{f}$ where $\hat{\mathbf{G}} = \text{diagonal}(\hat{g_1}, ..., \hat{g_n})$ a matrix with spectral factors on the diagonal representing the filter with $\mathbf{f}'$ as filter output. This architecture has high computational complexity $\mathscr{O}(n^2)$ for calculating the forward and inverse Fourier transformations, additional complexity of $\mathscr{O}(n)$ is required for parameters in each layer. The paper notes an additional issue that there is no guarantee of spacial localization of the filters.

MeshGAN develops a similar architecture to BEGAN [3]

In order to reduce the computational complexity the paper makes use of the work done in "Convolutional neural networks on graphs with fast localized spectral filtering" [12] and considers the spectral convolutions with polynomial filters in the Chebyshev basis. Formal the basis is expressed:

$$\tau_\theta(\lambda) = \sum_{j=0}^{p} \theta_j T_j(\lambda) \qquad T_j(\lambda) = 2\lambda T_{j-1}(\lambda) - T_{j-2}(\lambda)$$

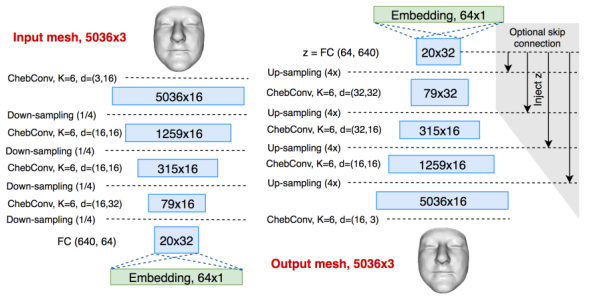
T_j is the Chebyshev polynomial which can be computed by

applying powers of the Laplacian to the feature vector as follows:

$$\mathbf{f}' = \Phi \sum_{j=0}^{p} \theta_j T_j(\tilde{\mathbf{\Lambda}}) \Phi^T \mathbf{A}\mathbf{f} = \sum_{j=0}^{p} \theta_j T_j(\tilde{\mathbf{\Delta}})\mathbf{f}$$

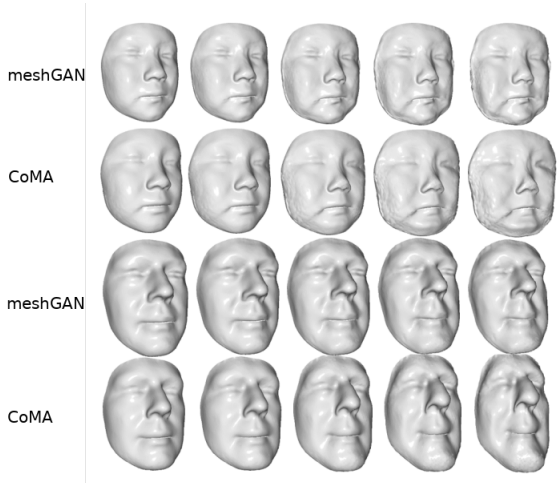
Where $\tilde{\Delta} = 2\lambda_n^{-1}\Delta - \mathbf{I}$ and $\tilde{\Lambda} = 2\lambda_n^{-1}\Delta - \mathbf{I}$. Then since the mesh is sparsely connected the computational complexity of the spectral convolutional network drops to $\mathcal{O}(n)$.

Figure 11: Network architecture of MeshGAN [10]



Now we can define the architecture of MeshGAN shown in figure 10. The encoder and generator are constructed with 4 Chebyshev convolutional filters with $K = 6$ polynomials for the encoder. After each convolutional layer the team uses the exponential linear unit (ELU) [11] to allow for negative activation's. Typical facial mesh datasets are high in detail and need to be down sampled, for this the team uses surface simplification which minimises quadratic error[16]. After embedding, the model needs to be upsampled, the team uses the barycentic coordinated of contracted vertices in the mesh described in [33]. Downsampling and upsampling are in 4 steps each changing the number of vertices by a factor of 4. At each upsampling step the team injects the latent vector $\mathbf{z}$ in order to encorage more facial detail development. For an optimiser the team uses a momentum based optimiser which should ensure any poor local optimisation minimums are ignored. The teams training hyperparameters are learning rate being 0.008 and decay rate being 0.99 over 300 epochs.

Figure 12: MeshGAN results [10]



Method	Generalisation	Specificity	FID
CoMA-ID	$\mathbf{0.442 \pm 0.116}$	1.60 ± 0.22	14.24
MeshGAN-ID	0.465 ± 0.189	$\mathbf{1.433 \pm 0.14}$	**10.82**

Table 2: Showing Specificity, Generalisation and FID for CoMA and MeshGAN on identity (lower is better)

Method	Generalisation	Specificity	FID
CoMA-E	0.606 ± 0.203	1.899 ± 0.27	22.43
MeshGAN-E	$\mathbf{0.605 \pm 0.264}$	$\mathbf{1.536 \pm 0.15}$	**13.59**

Table 3: Showing Specificity, Generalisation and FID for CoMA and MeshGAN on expression (lower is better)

We can see from figure 11 that MeshGAN produces results that are pleasing to the eye and free from the high frequency model distortions present in the CoMA samples. Additionally the figure shows that MeshGAN is able to control features accurately without changing identity in the process, this is most evident in the age manipulation in the last two samples.
Tables 2 and 3 show that MeshGAN performs well on quantitative metrics as well - out performing CoMA in all metrics for expression and all metrics but generalisation for identity manipulation. The poor generalisation results for both expression and identity show that there is room for improvement in terms of creating a model that is truly input agnostic. The source code for MeshGAN was not publicly released, however, there are open source implementations available.

9 Problems with GANs

9.1 Mode collapse

Mode collapse happens when the generator fails to optimise for the intended goal and rather all of the generated samples are very similar or even identical. The generator wins the adversarial game by creating a single data sample that always tricks the discriminator. The generator sacrifices the goal and settles on a degenerate equilibrium that satisfies the discriminator. This behavior is due to a poorly defined problem loss definition and can be solved by the WGAN architecture where the error distance is well defined [2].

9.2 Vanishing gradients

Deep neural networks such as MLPs and CNNs tend to loose information quickly as they train due to repeated gradients diminishing in size exponentially. This results in poor training performance as the gradient adjustments can be too small to make any significant difference, some architectures have been proposed to resolve the problem such as recurrent neural networks which pass information forward between layers to reduce the problem.

9.3 Convergence failure

Convergence failure is similar to mode collapse in that it is caused by a fundamentally flawed problem definition for the game between the generator and discriminator. The imbalance between the two agents causes the discriminators loss to rapidly decay settling to an equilibrium where samples

are too easily rejected while the generators loss does not converge as all. There have been many proposed solutions to the problem, however, there has not been a solution which does not significantly effect the training process. There has been promising on adding a weighted penalty function to the normal of the discriminator gradients which provides a separation from the generative process[34].

9.4 Generation output quality

Generally GANs are expensive to train on high resolution samples which is why many works function on low fidelity data. There have been many proposed solutions such as DC-GAN [32] which typically can produce higher quality results due to convolutional layers, ProGAN [24] which employs upsampling layers to enable far higher resolution results. However DC-GAN is still costly and ProGAN can produce checkerboard artefacts. In three dimensional works 3DMMs have been the standard for their statistical power and reduced computational cost when compared to working directly on point cloud like data.

10 Datasets

There are many openly available 3D face datasets that have been published, many of the best are covered in the paper Egger, Smith, Tewari, Wuhrer, Zollhoefer, Beeler, Bernard, Bolkart, Kortylewski, Romdhani, et al. by "3d morphable face models—past, present, and future". There are several common problems that can be seen in many of the datasets. There is a lack of consistency with the 3D facial scans provided, some include neck, ears, full rotation scan but others do not. The inconsistency may prove to be an obstacle for research as expensive preprocessing of the data will be needed to ensure consistency between different data samples within the training set.
There is a limited number of scans because creating a model is expensive and time consuming with many participants being needed as well as costly equipment and expertise. The low volume of data proves a difficulty as GANs and machine learning techniques in general require large datasets in order to perform well - the typical size is of $\mathcal{O}(10^2)$ whereas the typical 2D dataset is $\mathcal{O}(10^4)$.
The amount of information provided differs between datasets, some provide diffuse and specular albedo maps, hybrid normal maps, UV texture maps, depth maps, point clouds, triangle meshes and alignments in FLAME topology. Often with GANs the more information the model is provided the better however one much choose a dataset that provides enough valuable information to form a good understanding of a face as well as providing numerous high quality scans.

11 Conclusion

In this paper we have presented a basic explanation of the workings of a basic GAN and the base underpinnings of the technique. Introduced the problem of generating synthetic human faces both in two dimensions and three as well as the uses of such results. We have shown that in the literature there has been large emphasis on addressing the problems that arise when using GANs in addition to exploring such solutions. We emphasised some of the important work done in the area of three dimensional face generation and feature manipulation both in 2D and 3D domains using controllable GANs and compared their results. Further we presented the many representations of 3D faces and showed that there has been a focus on hybrid models that use a 3DMM in conjunction with novel deep learning techniques but there there has been little progress in the literature regarding direct application of deep learning GANs to a 3D model.
We discuss the various evaluation metrics that can be used to both quantitatively and qualitatively measure model performance and detect failure while comparing and contrasting the uses of such metrics. Using these metrics we compared the results of the three most relevant papers to controllable GANs when applied to 3D face generation.
The data quality, volume and availability is important when discussing an application of GANs which we expanded up and highlighted that there are many available datasets, however, their quality and consistency ranges widely.
In total we can see that 3D facial generation and manipulation is a highly valuable area of research that has many potential applications but still has much potential for improvement and modification as the majority of research focus has rested on 2D facial generation.

References

[1] David Barber Felix Agakov. 2004. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16, 320, 201.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Doina Precup and Yee Whye Teh, editors. Volume 70. PMLR, (June 2017), 214–223. `https://proceedings.mlr.press/v70/arjovsky17a.html`.

[3] David Berthelot, Thomas Schumm, and Luke Metz. 2017. Began: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*.

[4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*.

[5] Timo Bolkart and Stefanie Wuhrer. 2015. A groupwise multilinear correspondence optimization for 3d faces. In *Proceedings of the IEEE international conference on computer vision*, 3604–3612.

[6] Ali Borji. 2019. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179, 41–65.

[7] Alan Brunton, Augusto Salazar, Timo Bolkart, and Stefanie Wuhrer. 2014. Review of statistical shape spaces for 3d data with comparative analysis for human faces. *Computer Vision and Image Understanding*, 128, 1–17.

[8] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. 2018. To learn image super-resolution, use a gan to learn how to do image degradation first. In *Proceedings of the European conference on computer vision (ECCV)*, 185–200.

[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29.

[10] Shiyang Cheng, Michael Bronstein, Yuxiang Zhou, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. 2019. Meshgan: non-linear 3d morphable models of faces. *arXiv preprint arXiv:1903.10384*.

[11] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

[12] Michael Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

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

[14] 2010. *Surface representations for 3d face recognition.* (April 2010). ISBN: 978-953-307-060-5. DOI: 10.5772/8951.

[15] Farzan Farnia and Asuman Ozdaglar. 2020. Do gans always have nash equilibria? In *International Conference on Machine Learning*. PMLR, 3029–3039.

[16] Michael Garland and Paul S Heckbert. 1997. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 209–216.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. (2014). https://arxiv.org/abs/1406.2661.

[18] Zhongyi Han, Benzheng Wei, Ashley Mercado, Stephanie Leung, and Shuo Li. 2018. Spine-gan: semantic segmentation of multiple spinal structures. *Medical image analysis*, 50, 23–35.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

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

[21] Alec Jacobson and Olga Sorkine-Hornung. 2012. A cotangent laplacian for images as surfaces. *Technical Report/ETH Zurich, Department of Computer Science*, 757.

[22] Amina Kammoun, Rim Slama, Hedi Tabia, Tarek Ouni, and Mohmed Abid. 2022. Generative adversarial networks for face generation: a survey. *ACM Computing Surveys (CSUR)*.

[23] Rafal Karp and Zaneta Swiderska-Chadaj. 2021. Automatic generation of graphical game assets using gan. In *2021 7th International Conference on Computer Technology Applications*, 7–12.

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. (2017). https://arxiv.org/abs/1710.10196.

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

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

[27] Yann LeCun, Leon Botto, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition, (November 1998). http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf.

[28] Xiaoqiang Li, Liangbo Chen, Lu Wang, Pin Wu, and Weiqin Tong. 2018. Scgan: disentangled representation learning by adding similarity constraint on generative adversarial nets. *IEEE Access*, PP, (September 2018), 1–1. DOI: 10.1109/ACCESS.2018.2872695.

[29] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. Hologan: unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7588–7597.

[30] 2019. *Paired 3d model generation with conditional generative adversarial networks.* (January 2019).

[31] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. (2015). https://arxiv.org/abs/1511.06434.

[32] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. (2015). https://arxiv.org/abs/1511.06434.

[33] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. 2018. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 704–720.

[34] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. 2017. Stabilizing training of generative adversarial networks through regularization. *Advances in neural information processing systems*, 30.

[35] Divya Saxena and Jiannong Cao. 2021. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 54, 3, 1–42.

[36] Ygor Rebouças Serpa and Maria Andréia Formico Rodrigues. 2019. Towards machine-learning assisted asset generation for games: a study on pixel art sprite sheets. In *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. IEEE, 182–191.

[37] Sahil Sharma and Vijay Kumar. 2022. 3d face reconstruction in deep learning era: a survey. *Archives of Computational Methods in Engineering*, 1–33.

[38] Sahil Sharma and Vijay Kumar. 2020. Voxel-based 3d face reconstruction and its application to face recognition using sequential deep learning. https://link.springer.com/article/10.1007/s11042-020-08688-x#citeas.

[39] Lucas Terissi, Mauricio Cerda, Juan Gómez, Nancy Hitschfeld, and Bernard Girau. 2013. A comprehensive system for facial animation of generic 3d head models driven by speech. *EURASIP Journal on Audio, Speech, and Music Processing*, 2013, (February 2013), 5. DOI: 10.1186/1687-4722-2013-5.

[40] Mukhiddin Toshpulatov, Wookey Lee, and Suan Lee. 2021. Generative adversarial networks and their application to 3d face generation: a survey. *Image and Vision Computing*, 108, 104119.

[41] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-domain gan inversion for real image editing. In *European conference on computer vision*. Springer, 592–608.