# Expression Controllable 3D Point Cloud GAN (PCE-GAN)

Liam Watson
liam@watson.org.za
University of Cape Town
Cape Town, Western Cape, South Africa

## ABSTRACT

GANs are currently considered to be the best performing method for generation of visual content. Many recent works have demonstrated outstanding results when generating feature controlled images of human faces, however, generation of 3D facial data has not been able to obtain similar performance. The majority of works have focused on volumetric or spectral convolutions. In this paper we propose a novel method for generation of 3D faces directly operating on point sets named PCE-GAN. We describe the underpinning concepts of the work as well as the current state of the art models used for 3D face generation. We explain the architecture of PCE-GAN and provide both qualitative and quantitative results demonstrating that PCE-GAN can be used to generate high fidelity 3D face point clouds with temporally consistent, identity disentangled expression manipulation.

## CCS CONCEPTS

• **Generative Adveserial Network** → **3D Face Point Cloud Generation**; • **PointCloud** → *Surface Reconstruction*; • **GAN evaluation metrics** → FID,KID; • **GAN** → training stability.

## KEYWORDS

neural networks, face generation, classification, controllable GAN, datasets, point cloud, surface reconstruction

## 1 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) [11]. There has been remarkable progress using GANs on 2D images for arbitrary face generation and feature manipulation [19]. 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 [17]. Recently methods have been proposed with promising results working purely in the three dimensional domain such as MeshGAN [7] and CoMA [28]. Historically the large majority of 3D face research has focused on a hybrid learning architecture combining 3D Morphable Models (3DMMs) in conjunction with another deep learning technique [31].

### 1.1 Face Generation

There have been many works on face generation but most focus on 2D image data with emphasis on either feature control or output fidelity [1, 6, 12, 18, 19, 21, 22, 26].
Deep learning with 3D data has been demonstrated with many works focusing on classification with point clouds [25, 35] or a voxel based approach [5, 34].
For generation of 3D faces the majority of research has used a hybrid architecture combining either a GAN and 3DMM or AE and 3DMM [32] but some recent works have made remarkable progress operating directly on triangular meshes using spectral convolutions [7, 28]. The proposed solution in this paper seeks to circumvent the limitations of the hybrid architecture and spectral convolution by applying a GAN directly to point set facial data.

## 2 PROBLEM FORMULATION

In this work we develop a model that directly consumes an unordered point set as input. We define a point cloud as the set of $n$ points in 3D space $P = \{P_i | i = 1, ..., n\}$ where each $P_i$ is a vector $(x, y, z)$.
For face generation the input point cloud is directly sampled in both a classifier and GAN architecture. The proposed resulting controllable GAN consumes a controlled latent noise vector to produce a feature controlled point cloud $P$.

### 2.1 Deep Learning on Point Sets

Point Cloud data can be interpreted as a point set. Given a set of points in Euclidean space, the set has the following properties which can be leveraged in deep learning applications.

- Unordered. The ordering of points in the set does not encode any information - this is in opposition to discretized standards like 2D pixel arrays or 3D voxel grids.
- Metric space. The given point set forms a metric space with a Euclidean distance metric. This means that points are not isolated - neighboring points form a meaningful cluster that can encode macroscopic features in set. Models processing point cloud information must be able to capture the information in such local structures.
- Transformation invariance. Representations of the point set are invariant under affine transformations and as such any learnt representation should be as well.

## 3 RELATED WORK

In order to understand the significance of this work we must cover the techniques which have relevance both in terms of informing the architecture of PCE-GAN or for result comparison. In this section we describe the relevant works and how they differ with the solution proposed in this paper.

### 3.1 3D Morphable Model (3DMM)

The most successful face generation strategies up to date have used a statistical principle component analysis approach for linear combination face registration technique called a 3DMM. The model

is fundamentally limited by the type and amount of training data, control and structure. Recent techniques have attempted to solve the control problem by preceding the 3DMM with a machine learning model which learns how to tweak 3DMM parameters for a desired result. All faces produced by a 3DMM will be a linear combination of the faces registered to the model.

## 3.2 GANs

*3.2.1 GAN Introduction.* GANs were developed by Ian Good Fellow in 2014, they noticed that discriminative models had made rapid progress but deep generative models having made little progress due difficulty approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies as well as difficulty leveraging the benefits of piecewise linear units in the generative context [12]. 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 improving 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:

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

*3.2.2 DCGAN.* 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 - rather than processing linear data, the network accepts tensor data. CNNs have been demonstrated 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 [20]. 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 [26].

## 3.3 Controllable GANs

*3.3.1 Latent space.* An important concept in GANs and particularly controllable GANs is latent space which is a feature space where items which closely resemble each other have a shorter distance between each other. In GANs we can interpret the input space of the generator as a latent space hence we refer to the input vector as the latent vector. This interpretation of the input can be leveraged in order to isolate and disentangle features or effectively generate similar output by intelligently moving the latent vector within the latent space [27].

*3.3.2 Controllable GAN formulation.* A typical GAN has no control over the output features as we randomly select a point in the latent space as input for the generator to decode. A proposed technique for adding control is the conditional GAN which adds an additional input vector to the generator which encodes a class. This strategy encourages the generator during training to learn a disentangled representation of feature classes which can then be used for class specific generation. This approach essentially divides the latent space into class specific regions as class vectors and latent vectors are concatinated before being given to the generator. The main drawback of conditional GANs is that they require detailed labeling of training data and produce a discrete result - controllable GANs seek to circumvent these restraints while still retaining feature control. The strategy for feature control in a controllable GAN is to tweak the input latent vector to intelligently locate the desired position in the latent space that will decode to a desired feature set.

Controllable GANs attempt controlled latent space exploration for feature controlled generation while still maintaining unsupervised training. The normative structure consists of a pretrained classifier and GAN with the classifier gradients being used to inform latent vector changes to achieve a desired feature in the output - many new techniques incorporate some disentanglement scheme during training to ensure that latent space dimensions are well correlated to a certain feature. The feature manipulation process is iterative where a face is generated and a classifier probability is obtained for the synthetic face. The latent vector is then updated as $z_i = z_{i-1} + l\nabla c_e$ where $\nabla C_e$ is the gradient of the classifiers prediction of a requested expression $e$ and $l$ is an update factor much like a learning rate.

## 3.4 MeshGAN

MeshGAN is a variant of BEGAN [2] that learns a non-linear 3DMM directly from 3D mesh facial data [7]. MeshGAN works directly on embeddings of manifold triangular meshes. Using a discretized Laplacian operator the paper uses spectral mesh convolutions to define a graph convolutional neural network with the spectral convolution operation defined as $\mathbf{f}' = \Phi\hat{\mathbf{G}}\Phi^T\mathbf{f}$ where $\hat{\mathbf{G}}$ is a diagonal matrix of spectral factors. This convolutional operation has computational complexity $O(n^2)$ due to Fourier decomposition so the paper employs Chebyshev convolutional filters introduced in ChebNet [9] to reduce computational complexity to $O(n)$. This approach enables direct operation on the triangular meshes but introduces two sources of error - the spectral decomposition of the face mesh introduces finite Fourier transform error and the introduction of Chebyshev polynomials introduces Chebyshev polynomial approximation error. In this paper we propose circumventing the error introduced by Chebyshev convolutional filters and spectral decomposition by operating directly on the facial point set data [9, 14].

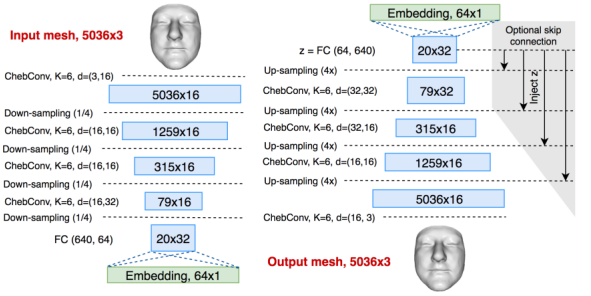


**Figure 1: Network architecture of MeshGAN [7].**

Given the above explanation we can define the architecture of MeshGAN displayed in Figure 1. 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) [8] 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[10]. After embedding, the model needs to be upsampled, the team uses the barycentic coordinated of contracted vertices in the mesh described in [28]. 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 **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.

## 3.5 CoMA

Convolutional Mesh Autoencoders (CoMA) was the first paper to make use of spectral decomposition and Chebyshev convolutional filters applied to the a face mesh - treating the mesh as a connected graph. The CoMA paper introduces the standard dataset used in 3D face research as well as a preprocessing scheme which was adapted for this work. The CoMA autoencoder architecture is displayed in Figure 2 where each block refers to a Chebyshev convolutional filter with $K = 6$ Chebyshev polynomials and arrows refer to up and down sampling.

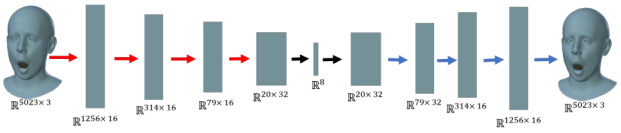


**Figure 2: Showing the architecture for CoMA [28].**

## 3.6 PointNet

PointNet [25] is a deep learning architecture for point cloud classification and segmentation - the model proposed in the paper is generally considered to be the best performing point cloud classifier. The model leverages the properties of point sets to inform its architecture using a symmetry function on input, local and global information aggregation and a joint alignment network - such a symmetry function can only be used while point set properties hold.

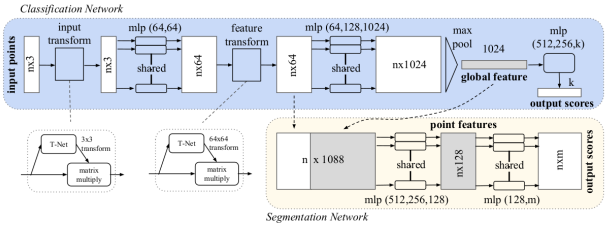


**Figure 3: PointNet Architecture showing model composition for classification and segmentation [25].**

Point set data contains sufficient structural information for a model to learn distinctive features while having a dramatically lower computational cost when compared to spectral or volumetric approaches. Additionally direct convolution of point set information does not suffer from spectral error introduced when operating directly on graph meshes using spectral convolutions.

# 4 PCE-GAN

We introduce PCE-GAN which is a controllable GAN variant that makes use of a deep convolutional GAN architecture and PointNet as a classifier.
The generator is given a random noise vector $z$ and produces a pointcloud face $P_{face}$.
The discriminator is given a point set face $P_{face}$ and produces a probability of a face being real or fake.
The classifier is given a point set face $P_{face}$ and asked to assign probabilities to a set of expression classes $C_{exp}$.
The generator and classifier are separately trained on a set of expression point clouds for various identities $P_E$. The GAN training dataset includes all time series intermediate expressions in order for the generator to learn interpolations between various facial expressions using intermediate expressions. After training we iteratively obtain classifier class probabilities from an arbitrary generated face $P_i$ using initial noise vector $z_i$. The gradient of the classifier is then used to update $z_i$ in order to increase the probability of some intended class $c_e \in C_{exp}$. In this way we can intelligently move the latent vector $z$ within the latent space to generate the intended point cloud face expression as well as intermediate expressions. The architecture for the GAN (generator and discriminator) and PC-GAN are displayed in Figures 4 and 5 respectively. The GAN leverages the architecture introduced by DCGAN [26] with 1D convolutional layers where pointset information is linearly encoded in $(x, y, z)$ channels which can be viewed as analogous to a 1D RGB encoded image.

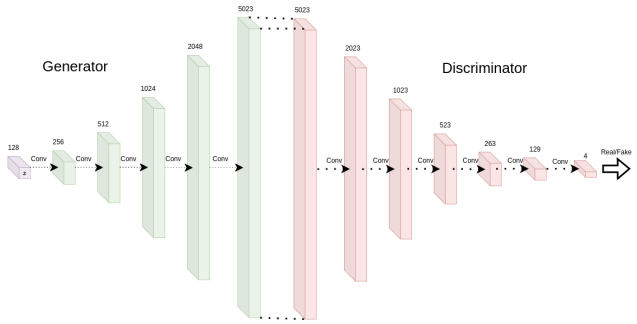


**Figure 4: Showing the generator and discriminator used in PCE-GAN.**

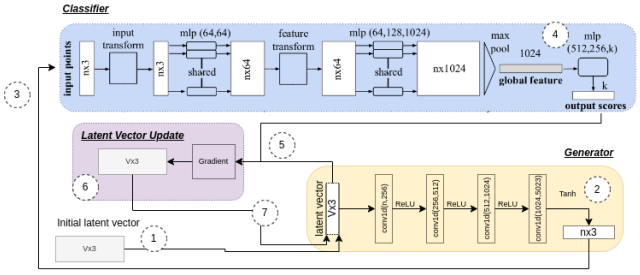


**Figure 5: Showing the controllable generation process and architecture used in PCE-GAN. Numbers 1 to 7 indicate process steps.**

The hyperparameters used for the GAN and classifier training are presented in tables 1 and 2 respectively. Hyperparamter tuning was completed using wandb [3], informed by loss metrics, qualitative assessment and hardware limitations. Importantly the noise dimension must be large enough to define a latent space that can capture all information about identity and expression while remaining computationally reasonable. The expression count of 4 was chosen to fully test the method (binary classification is typically trivial). The expressions chosen were: bare teeth, cheeks in, mouth open and high smile as they give a good balance between facial structure changes in bone and skin.

**Table 1: Hyperparameter for PCE-GAN generator.**

| Hyperparameter | Value |
|---|---|
| learning rate | $1 \cdot 10^{-5}$ |
| batch size | 10 |
| Epochs | 2000 |
| Data samples | 1000 |
| Expression Count | 4 |
| Noise dimension | 128 |
| Cost function | BCE Loss |
| Optimiser | Adam |

**Table 2: Hyperparamters for PCE-GAN classifier.**

| Hyperparameter | Value |
|---|---|
| learning rate | $1 \cdot 10^{-5}$ |
| batch size | 6 |
| Epochs | 1000 |
| Data samples | 1000 |
| Expression Count | 4 |
| Cost function | NLL Loss |
| Optimiser | Adam |

# 5 EVALUATION STRATEGIES

## 5.1 Data

There are numerous datasets which are available containing various types of facial data. For this work we will use the most commonly explored in the literature to enable comparison of results as well as ease of development.

The CoMA dataset contains triangle meshes (80K–140K vertices) of the full head including face, neck and ears of 12 individuals in 12 extreme expression sequences.

In this paper we use the open source CoMA preprocessing to down sample the high resolution models and produce point sets of 5023 elements [28]. The down sampled point sets can be seen in Figure 6. Down sampling is required as in the original faces there are many redundant points which would increase the computational cost and memory requirements dramatically without contributing a significant amount of information. For structural information 5023 is a sufficient quantity to capture macroscopic features such as eyes, ears, nose, mouth, bone structure and chin size. High frequency detail is typically not captured directly in mesh data as this would require on the order of $10^6$ vertices or more and as such is typically encoded in texture based approaches.

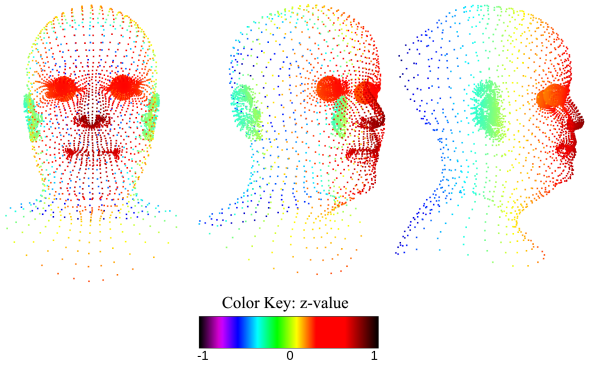


**Figure 6: Showing a data sample point set after CoMA preprocessing from different angles where point color shows z-depth.**

## 5.2 Training and Testing Pipelines

The pipeline can be broken up into discreet components displayed in Figures 8[1] and 7 [2]. The facial information is split into a test dataset and training dataset. To ensure that the facial data is of the correct format ready for model consumption a preprocessing step is required which includes extraction of point set information and down sampling as GANs have high memory and computational requirements. A data loader scheme is needed to manage batching, labels and feeding of data to the neural network. In order to track training progress, performance, ensure consistency and enable efficient hyperparameter tuning a robust training framework is needed. A robust model saving scheme is included to ensure that model training progress can be tracked historically as well as to ensure that training may be resumed in the case of hardware failure.

For a trained model we require a robust and consistent evaluation framework which enables both quantitative and qualitative metric gathering. This pipeline will consume a model and testing data in order to produce such metrics. For quantitative metrics the pipeline will produce FID, KID, generalisation and specificity. For qualitative metrics the pipeline will cycle through all generated faces in both point cloud and mesh visualisations.

Both the training and testing pipelines are general enough to work in a plug and play fashion - working interoperability between models. Future works may use the pipelines for comparison work or other such 3D research projects.

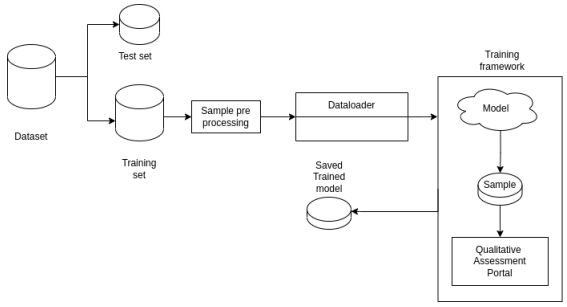


**Figure 7: Diagram showing the pipeline used for training PCE-GAN.**

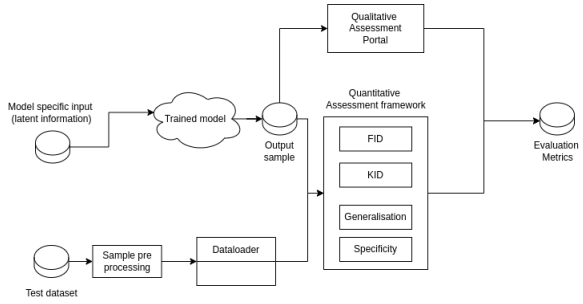


**Figure 8: Diagram showing the pipeline used for evaluation of PCE-GAN.**

[1]The quantitative pipeline was majority implemented by L. van den Handle
[2]The training framework was adapted by that constructed by S. Oliver.

## 5.3 Comparison Methods and Models

For comparison models we will use MeshGAN by Cheng et al. [7] and CoMA autoencoder [28] as they use the CoMA dataset and are the best performing works to date. For consistency we will additionally be testing the generator from PCE-GAN without latent manipulation to isolate control effects - referred to as DCGAN.

*5.3.1 Quantitative.* In this paper we use FID and KID to evaluate PCE-GAN as they are the typical metrics used in GAN research for quantitative assessment as well as generalisation and specificity as introduced in meshGAN. **FID** (Frechet Inception Distance) [15] is the most commonly used metric for GAN evaluation but has recently been superseded by KID. FID contrasts the feature extracted distribution of synthetic images with the distribution of real images used to train the generator. For feature extraction we use the inception network v3. **KID**(Kernel Inception Distance) [4] is an unbiased metric that computes the squared maximum mean discrepancy between feature representations of the real and generated images. For both FID and KID the point cloud must be converted into a 2D image. For this Poisson Surface reconstruction is used and then a full frontal rasterization is created with Phong shading as this is typical for 3D research. It should be noted that this process is highly hyperparameter dependent with point set normal estimation, Poisson surface reconstruction parameters and smoothing techniques.

**Generalisation** measures the models ability to represent unseen faces that were not encountered during training. Computation of generalisation error is done by per-vertex Euclidean distance between samples in the test set $\mathbf{x}$ and generated samples $\mathbf{x}^*$ which is expressed $\mathbf{x}^* = \text{argmin}_z |\mathbf{x} - G(\mathbf{z})|$. The overall metric is represented as a mean and standard deviation over all faces in the testing set.

**Specificity** evaluates the validity and realism of face reproduction by a generator. A set generated faces are synthesised and proximity to real faces in the test set is calculated by per vertex euclidean distance as in generalisation. The metric is represented as the mean and standard deviation of the proximities of each generated face to its closest test set face.

*5.3.2 Qualitative.* Qualitative evaluation is 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 [30]. Qualitative metrics can additionally be important for detecting training performance and failure such as over fitting or mode collapse. In this paper results will be shown for the reader to perform their own analysis in terms of preference judgement and mode collapse.

## 5.4 Surface Reconstruction

Point cloud surface reconstruction is the process of converting a point set to a mesh object. There are a variety of techniques that have been proposed such as alpha shapes, ball pivoting and Poisson reconstruction. For this work we rely as Poisson as it produces closed meshes of high quality. Poisson surface reconstruction relies heavily on the quality of point set normal's which we must recover in this paper by normal estimation which is a highly parameter sensitive operation.

## 6 EXPERIMENTS

Experiments for PCE-GAN are split into three parts. First we show PCE-GAN can be used to synthesize human faces. Second we show PCE-GAN can be used to generate continuously variable expressions. Thirdly we analyze PCE-GANs learnt latent space, stability and complexity.

### 6.1 Experiment 1: Face Generation

In this section we describe the ability for GANs to produce face point clouds.

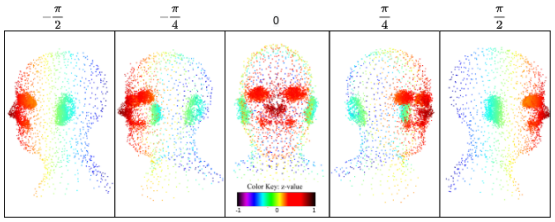


**Figure 9: Point cloud generated by PCE-GAN with arbitrary latent vector from different camera rotation angles. Point color shows z-depth.**

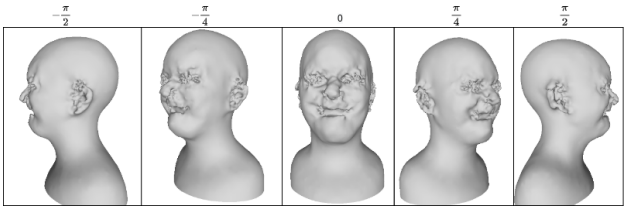


**Figure 10: Surface reconstruction of point cloud generated by PCE-GAN with arbitrary latent vector.**

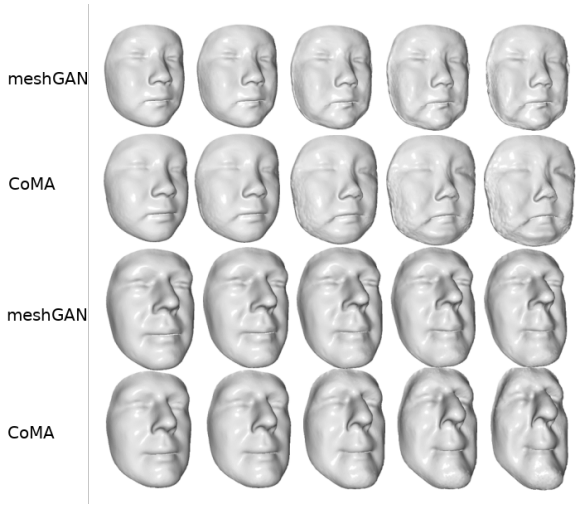


**Figure 11: MeshGAN and CoMA on identity [7].**

We can see from Figure 9 that PCE-GAN is able to stably generate recognisable human face point clouds. The figure shows that the generated faces have the correct point set densities that we expect for a face, with clusters forming facial features: eyes, ears, nose and mouth. Figure 10 shows us that the generated faces capture all the required features, however, the noise in the generated output in combination with the inaccuracy of vertex normal estimation and surface reconstruction causes macroscopic mesh aberrations clearly visible on facial structures. The large macroscopic aberrations caused by spurious points in PCE-GAN are in contrast to the high frequency aberrations in the meshGAN and CoMA which are caused by amplified error terms in the spectral convolutions both architectures employ. The final mesh that CoMA and meshGAN produce contains a high degree of topological and facial consistency which is due to the use of spectral convolution to directly encode mesh data where as PCE-GAN must recover this information via normal estimation and surface reconstruction.

The noise in PCE-GAN output inherently limits the quality of output results dramatically, especially when applying a surface reconstruction technique. The source of the noise is due to a GANs inherent structure and the problem definition - a point is considered spurious if its position is off by just $1e^{-3}$ or less. Many techniques were attempted in this paper to reduce the noise. It was proposed that batch normalisation would aggregate faces and hence encourage points to lie in a surface, however, batch moralisation causes training instability. The possibility of empowering the discriminator to force the generator to produce points on a surface as in the training data, this technique has promise, however, gives rise to mode collapse as the minimax game posed in a GAN becomes unbalanced. Methods that did improve noise reduction are learning rate decay, using the Adam optimiser and longer training times. This indicates that optimisation is likely a key factor in reducing noise such as the complexity of the model increasing modelling power, learning rate decay over longer training times for fine tuning the point placements to refine the facial structure and reduced aberrant points and a well constructed optimization algorithm to ensure stable training.

**Table 3: Showing Specificity, Generalisation and FID for CoMA and MeshGAN on identity from MeshGAN paper [7] and PCE-GAN without feature manipulation (lower is better).**

| Method | Generalisation (mm) | Specificity (mm) | FID |
|---|---|---|---|
| CoMA-ID | 0.442 ± 0.116 | 1.60 ± 0.22 | 14.24 |
| MeshGAN-ID | 0.465 ± 0.189 | 1.433 ± 0.14 | **10.82** |
| PCE-GAN | **0.747 ± 0.0163** | **0.809 ± 0.0334** | 13.27 |

When we look at the quantitative metrics in Table 3, we see that PCE-GAN outperforms CoMA and meshGAN in specificity with standard deviation over 200 samples. The relatively poor generalisation score shows that PCE-GAN is not as good at representing unseen faces as CoMA and MeshGAN. This is due to three factors namely architecture, data and training. The spectral convolutions and mesh data used in previous works encode more information about a face which can enable such architectures to learn a more general idea of a face including information on vertices and

faces. Additionally, MeshGAN and CoMA have used larger training datasets which exposes the models to a higher variety of face types and expressions. Compared to the simplicity of a 3DMM which has even less variation we can see that embedding information about the idea of a face rather than a linear combination of already seen faces used in 3DMMs gives rise to more diversity in results. The low specificity value shows that faces generated by PCE-GAN have high validity and are likely to be considered highly face like by a human evaluator. Again PCE-GAN outperforms CoMA but not meshGAN on FID which demonstrates that generated faces are of high quality and diversity. It is likely that the poor surface reconstruction is the reason for meshGANs better performance on FID. The reason for such a good specificity score is likely due to the point set data being a more easily learnt data format than a spectral graph mesh.

## 6.2 Experiment 2: Expression Manipulation

In the following experiment we detail PCE-GANs ability to control generated output while retaining generation quality and stability.

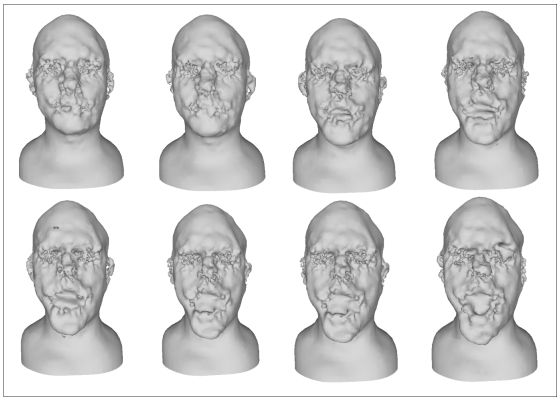

**Figure 12: PCE-Expression manipulation for "open mouth" from baseline face (top left) to 99.99% classifier confidence (bottom right).**

**Figure 13: Showing PCE-GAN generated "open mouth" face with surface reconstruction tuning and removed aberrant points.**

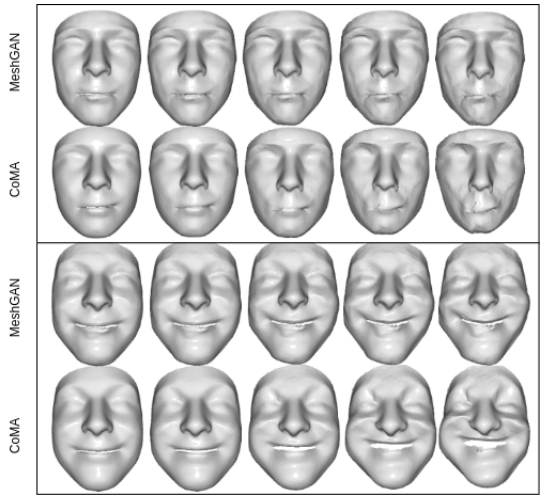


**Figure 14: MeshGAN and CoMA Expression manipulation results [7].**

From Figure 12 we can see that PCE-GAN is able to produce expression controlled point cloud facial data. The figure shows us that PCE-GAN expression control produces facial structure consistent results over iterations - as the mouth progressively opens we see the chin and jaw move down, skin begin to stretch over facial bones, mouth width decrease and top lip move up while all structurally unrelated facial features remain constant. However there are some higher frequency unwanted changes in the eyes and nose that are caused by mesh reconstruction errors, increased by spurious points as without latent vector manipulation. It can be seen that latent vector manipulation does not increase the quantity of spurious points, in fact it is likely to reduce them given a well trained classifier as latent manipulations should increase the classifiers expression score which is likely to involve point cloud smoothing. Additionally one can see that results are temporally consistent which is evident in that transition expressions are valid - in the opening mouth Figure 12 we see that all facial features remain the same with the exception of the mouth slowly opening. The temporal consistency is likely due to the inclusion of the time series data in the training set and is a good indication that PCE-GAN has good structural knowledge of a face.

When comparing PCE-GAN feature manipulated results to those of meshGAN and CoMA we can see that PCE-GAN results contain much more aberrant data. However, in the previous works, the more the expression is manipulated the larger the error from the spectral convolution grows - which can be seen in the incremental increase of high frequency noise in the generated meshes. It should be noted that, although not presented CoMA and meshGAN expression manipulation can introduce identity entanglement issues, these do not appear to be present in PCE-GAN.

As a further demonstration of the impact that surface reconstruction and aberrant points have on the resulting mesh we include Figure 13. The mesh displayed was created by hand removing aberrant points, tuning normal estimation, surface reconstruction parameters and adding eye balls which are a particular difficulty for the surface reconstruction algorithms.

**Table 4: Showing Specificity, Generalisation and FID for CoMA, MeshGAN from MeshGAN paper [7] and PCE-GAN on expression (lower is better).**

| Method | Generalisation (mm) | Specificity (mm) | FID |
|---|---|---|---|
| CoMA-E | 0.606 ± 0.203 | 1.899 ± 0.27 | 22.43 |
| MeshGAN-E | **0.605 ± 0.264** | 1.536 ± 0.15 | 13.59 |
| PCE-GAN | 0.659 ± 0.067 | **0.614 ± 0.129** | **12.06** |

The results found for PCE-GAN in Table 4 were obtained by generating 200 faces with expression set to the classifiers initial best class. Such synthetic faces are then used to compute the relative metrics by comparing to faces in the test dataset. For these quantitative results we can see that control influenced PCE-GAN improves upon previous results, showing, improvements in generalisation, specificity, and FID. This property of the controllable architecture is likely due to classifier gradients informing a reduction in spurious points as such points are opposed to classifier expression expectation. Interestingly we do see an increase in standard deviation which indicates that the quality and variety of PCE-GAN generated faces varies more when control is introduced - such behavior is due to classifier biases.

**Table 5: Showing KID, FID, Generalisation and Specificity for PCE-GAN on 4 expressions (lower is better).**

| Expression | KID | FID | Generalisation | Specificity |
|---|---|---|---|---|
| Bare teeth | 0.87 | 16.59 | 1.161 ± 0.157 | 0.8788 ± 0.2698 |
| Cheeks in | 0.63 | 15.48 | 1.329 ± 0.006 | 1.162 ± 0.384 |
| Mouth Open | 0.32 | 13.03 | 0.957 ± 0.148 | 0.7610 ± 0.1662 |
| High Smile | 0.59 | 15.05 | 0.7771 ± 0.0198 | 1.032 ± 0.388 |

Table 5 shows us the model performance between expressions. We can see that for more complex expressions such as cheeks in and bare teeth the model performance decreases, however, still out performing CoMA. For cheeks in and high smile we see poor specificity scores - this result is due to the expressions consisting of smaller structural differences. PCE-GAN likely does not capture this detail well with the 5023 point resolution.
It is critical that the classifier has a good understanding of facial structure for well informed movement of the latent vector. We have demonstrated that our classifier is sufficiently able to control movement within the latent space for a desired expression, however, it is important to analyse the classifier for an understanding of its control influence. To this end we include the confusion matrix in Figure 15. The accuracy of the classifier is relatively low which can be seen. For the purposes of this work a high accuracy - although desirable - is not entirely relevant as we are not interested in absolute predictions but slight gradient adjustments. In fact a high accuracy may lead to poor mode collapse like results where facial expressions are too tightly bound. Figure 15 shows that the classifier confuses structurally similar expressions such as mouth open and bare teeth, high smile and cheeks in, as well as high smile and bare teeth. The expression confusion behavior is expected as the time series data can cause structural overlap of expressions. An important case is that of mouth open where the classifier gets the most incorrect predictions, but, is still able to suggest suitable latent vector updates to achieve the desired mouth open expression.
Another finding from Table 5 is that overall quantitative results worsen when we impose an expression on a generated face. We can see this when comparing PCE-GAN metrics between Tables 5 and 14 where overall Table 5 shows worse scores in all metrics. This behavior is due to expression control being imposed on initial faces where the classifier struggles to find effective latent vector changes to improve the expression score. Such issues would be improved by including expression to expression rather than neutral to expression time series data.

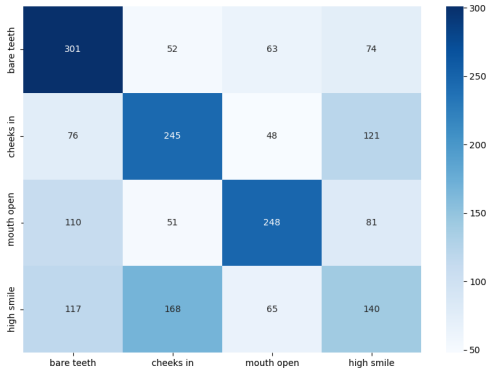


**Figure 15: Showing confusion matrix for PointNet classifier.**

## 6.3 Experiment 3: PCE-GAN analysis

In this section we discuss PCE-GANs learnt latent space, stability, time complexity and memory usage.

*6.3.1 Learnt Latent Space.* It is hard to explicitly visualise a latent space with high dimensionality (128) as such we use random generation as a proxy to explicit latent space mapping. From Figure 16 below we can see that the latent space consists of varying faces and expressions. It should be noted that although the orientation of faces remains constant, size does not which is an indication that the latent vector encodes generated face scale as well as identity and expression. This is a positive results as it indicates that both identity, size and facial expression are learnt by the generator but latent vector manipulation is able to disentangle features - changing only expression while others remain constant. The latent space visualisation additionally demonstrates that a variety of identities are represented in the latent space with varying facial bone structures seen by differing jaw lines and cheek bones. Additionally we see varying facial textures with some being smooth and others showing ridges in the neck such as an adams apple or higher frequency deformations commonly seen in elderly individuals.
A diverse latent space is important as this enables a broad range of initial identity possibilities which is the central goal of such a generative architecture, GANs can suffer from latent diversity collapse where the input space gets mapped to a single output. The fact that PCE-GAN does not suffer mode collapse like this, demonstrates that point set information - although hard to mesh - is a suitable data for GAN training.

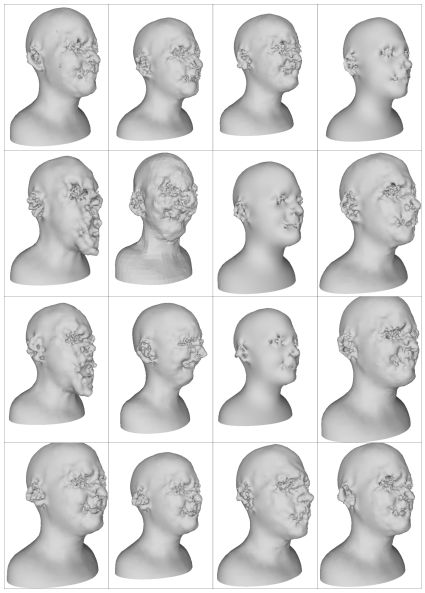

**Figure 16: Showing faces generated on random latent vectors.**

*6.3.2 Stability.* There are two main places to look for stability in PCE-GAN - training and feature manipulation. For training stability we seek a downward trend in face metrics during training which can be seen in Figure 17. For feature manipulation we seek to see a smooth classifier prediction score which can be seen in Figure 18.

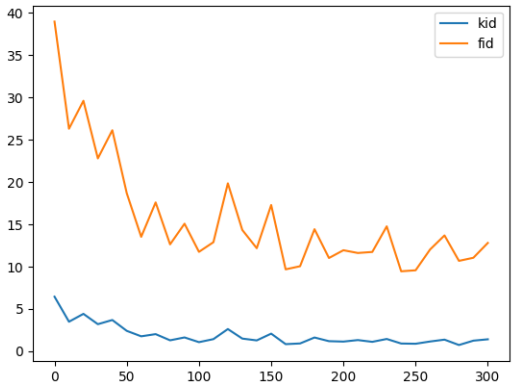


**Figure 17: Showing generator KID and FID over 300 epochs of GAN training.**

The above Figure 17 shows us that PCE-GAN is able to stably improve the quality of generated results over longer training intervals without exhibiting catastrophic failure modes such as vanishing gradients or convergence failure. Additionally we can see from Figure 16 that PCE-GAN does not exhibit mode collapse in that generated faces vary in identity and expression.

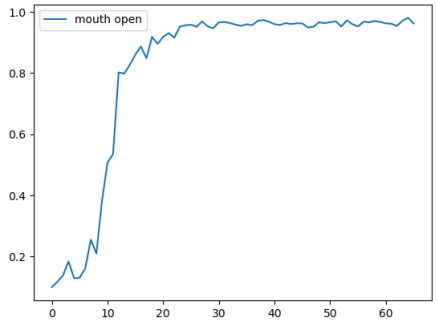


**Figure 18: Showing classifier confidence over successive latent vector manipulations for mouth open expression.**

Figure 18 demonstrates that during the control step of PCE-GAN classifier probabilities tend relatively smoothly to a certain prediction. The jagged lines seen in the figure are caused by a relatively large gradient update constant $l$ which can cause overly large adjustments to the latent vector.

*6.3.3 Time and Space complexity.* It is well known that the time complexity of convolutional layers is $O(n \cdot k \cdot d^2)$ where $n$ is the sequence length, $d$ is the representation dimension, $k$ is the kernel size [33]. For the GAN used in PCE-GAN $k = 1$, $d = 3$ and $n$ varies by layer but for large layers $n >> d$ so complexity reduces to $O(n)$. For the classifier PointNet claims time and space complexity of $O(n)$. So the total time and space complexity of PCE-GAN is $O(n)$.

## 7 DISCUSSION

It can be seen that PCE-GAN is able to produce synthetic point set facial data that performs well in both qualitative and quantitative metrics. However in general GANs have been proven to produce inherently noisy results which - in part - can be reasoned as enabling creativity. In images aberrant pixels can be seen as slight discontinuities in color or information, however, when imposing a surface reconstruction on a mesh with spurious points we find a highly distorted triangular mesh due to the inherent limitations of point set normal estimation and surface reconstruction algorithms ability to navigate spurious points. We have been able to demonstrate in figure 13 that with some human intervention spurious points can be removed as well as more detailed turning of point set normal estimation, surface reconstruction and mesh smoothing hyperparameters can be done. When such steps are taken we can recover a high fidelity face mesh. An expert in 3D modelling would be able to produce even better results given a base from PCE-GAN. For practical use the benefits of identity disentanglement and temporal consistency may prove highly valuable for industries that require a large variety of arbitrary facial identities along with expression animations such as video game characters.

We have seen that PCE-GAN generations have favorable FID and KID results which is a good indication of the quality and variety of results. It should be noted that such results are highly surface reconstruction and shading dependent and as such even better results could be obtained by manual tuning of surface reconstruction parameters to improve the face. Additionally a texture applied to

the surface would likely improve results as the IS v3 network is primarily trained on typical human faces which have texture as a feature.

Additional evidence that PCE-GAN produces results which are varied in identity is the favorable generalisation, indicating that PCE-GAN produces faces that are of high quality but varied in terms of identity.

The specificity results show that PCE-GAN produces faces that are well matched to those in the training set in terms of quality which indicates that one may be able to generate faces of good quality as long as the training dataset is of high quality. Generalisation and specificity are good measures for this work as they are independent of surface reconstruction. This means that they are a more accurate measure of the underlying model performance irrespective of post processing hyperparameters or technique.

It is important to note that PCE-GAN exhibits a high degree of stability in both training and control which is a good indication that the problem formulation is valid. GANs are highly sensitive to poor problem formulations and can often exhibit many generation and training failure modes such as mode collapse, vanishing or exploding gradients, failures to converge or even divergence - PCE-GAN exhibits none of these problems associated with GANs. To further support the motivation of point set data for face generation is the low computational and spacial complexity of PCE-GANs architecture which is not a given for works in the 3D domain.

# 8 CONCLUSIONS

In this paper we have demonstrated that point set data can be used for 3D face generation using GANs. We have proven that such a GAN can be used in combination with a classifier in a controllable GAN architecture to generate expression controlled faces without loss of quality or stability. Further it was demonstrated that the control step can improve the quality of synthetic faces. Generated faces from PCE-GAN have been proven to out perform previous works in some quantitative measures, expressing a high level of validity and variation. We have discussed the problems associated with PCE-GANs reliance on a processing surface reconstruction step that can produce undesirable meshes, however, such problems can be mitigated with little manual intervention and tuning. We have proven that PCE-GAN has a well defined and feature rich input latent space. Through experiments we have demonstrated that PCE-GAN is stable during training and feature manipulation. It was proven that the architecture of PCE-GAN is linear in both space and time complexity. Additionally, we have demonstrated that expression manipulation is temporally consistent and identity disentangled which means that intermediate expressions can be used as expression animations for a single character facial animations.

# 9 LIMITATIONS AND FUTURE WORK

## 9.1 Surface Reconstruction

Reconstruction of a mesh from the point set has proven to be a large flaw of the point set generation method. In further works, investigations into including point set normals in the training and generated results. This inclusion would enable both more detailed generation and accurate surface reconstruction.

## 9.2 GAN Improvements

The current formulation of PCE-GAN gives rise to spurious points in the generated point set. Further investigations into more advanced GAN architectures such as WGAN or WGAN-GP architectures may reduce the noisy results as such techniques have been proven to outperform DCGANs in most applications.

## 9.3 Training Data

For future work where detailed identity is important, higher fidelity training data is recommended as down sampling to 5023 points reduces the identity information dramatically while retaining the expression information. In order to increase the likelihood of high frequency detail (such as wrinkles) being included in generated results a higher number of training points is required. A larger training set will likely improve the quality of synthetic point sets, enabling a smaller learning rate - this does require more computational and memory resources. When extending the dataset one should include an increased variety of identities to broaden the controllable GANs ability to generate unseen faces. Additionally, the inclusion of expression to expression time series data would increase the models ability to produce temporally valid expressions and enable the model to better impose control over a variety of initial starting expressions.

## 9.4 Larger Degree of Feature Control

Future work may choose to increase the dimension of feature control from expression to identity features. Such an investigation would require a classifier trained on detailed face property labelling such as the size of facial expressions, age, gender, etc. Training data for such a model may be obtained by means of facial scans or use of manually constructed data that is often used in content creation.

## 9.5 Data Representations

It has not been proven that point set data is the most suitable representation for face generation. Further investigations may find an encoding of point set data more easily learnable by a GAN such as different coordinate systems which may emphasise relative point densities, embeddings of points into a space that emphasises a specific attribute of the data or a voxel approach to emphasise volumetric consistency. Different data representations my additionally aid surface reconstruction with volumetric information or a reduction of spurious points due to a higher degree of spacial accuracy.

## 9.6 Metrics

Future work may want to invest into phong shaded 3D face trained IS model for more easily comparable results.


# ACKNOWLEDGMENTS

Gracious thank you for the diligent and insightful supervision from Dr. D. Moodley.

Thank you to L. van den Handle and S. Oliver for their contribution to the experimental pipeline.

# REFERENCES


[1] 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, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 214–223. https://proceedings.mlr.press/v70/arjovsky17a.html

[2] David Berthelot, Thomas Schumm, and Luke Metz. 2017. Began: Boundary equilibrium generative adversarial networks. *arXiv:1703.10717* (2017).

[3] Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. https://www.wandb.com/ Software available from wandb.com.

[4] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. https://doi.org/10.48550/ARXIV.1801.01401

[5] Weiwei Cai, Dong Liu, Xin Ning, Chen Wang, and Guojie Xie. 2021. Voxel-based three-view hybrid parallel network for 3D object classification. *Displays* 69 (2021), 102076.

[6] 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 (2016).

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

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

[9] 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 (2016).

[10] 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.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

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

[13] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[14] Mingguo He, Zhewei Wei, and Ji-Rong Wen. 2022. Convolutional Neural Networks on Graphs with Chebyshev Approximation, Revisited. https://doi.org/10.48550/ARXIV.2202.03580

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

[16] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55

[17] Amina Kammoun, Rim Slama, Hedi Tabia, Tarek Ouni, and Mohmed Abid. 2022. Generative Adversarial Networks for face generation: A survey. *ACM Computing Surveys (CSUR)* (2022).

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation. https://arxiv.org/abs/1710.10196

[19] 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.

[20] 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

[21] Minhyeok Lee and Junhee Seok. 2019. Controllable Generative Adversarial Network. *IEEE Access* 7 (2019), 28158–28169. https://doi.org/10.1109/ACCESS.2019.2899108

[22] 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.

[23] The pandas development team. 2020. *pandas-dev/pandas: Pandas*. https://doi.org/10.5281/zenodo.3509134

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *CoRR* abs/1912.01703 (2019). arXiv:1912.01703 http://arxiv.org/abs/1912.01703

[25] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2016. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. https://doi.org/10.48550/ARXIV.1612.00593

[26] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. https://arxiv.org/abs/1511.06434

[27] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. https://arxiv.org/abs/1511.06434

[28] 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.

[29] 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)*.

[30] Jun'ichiro Seyama and Ruth S Nagayama. 2007. The uncanny valley: Effect of realism on the impression of artificial human faces. *Presence* 16, 4 (2007), 337–351.

[31] Sahil Sharma and Vijay Kumar. 2022. 3D Face Reconstruction in Deep Learning Era: A Survey. *Archives of Computational Methods in Engineering* (2022), 1–33.

[32] 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 (2021), 104119.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. https://doi.org/10.48550/ARXIV.1706.03762

[34] Cheng Wang, Ming Cheng, Ferdous Sohel, Mohammed Bennamoun, and Jonathan Li. 2019. NormalNet: A voxel-based CNN for 3D object classification and retrieval. *Neurocomputing* 323 (2019), 139–147.

[35] Min Zhang, Yifan Wang, Pranav Kadam, Shan Liu, and C-C Jay Kuo. 2020. Pointhop++: A lightweight learning model on point sets for 3d classification. In *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3319–3323.

[36] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. 2018. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847* (2018).


# A SOFTWARE EXPLANATION

The implimentation of PCE-GAN and the associated training and testing pipelines was completed in python with the following libraries used:

(1) NumPy [13]
(2) PyTorch and Torchvision [24]
(3) Pandas [23]
(4) open3D [36]
(5) Matplot lib [16]

The CoMA autoencoder is also open-source and publicly available so it will be used as a baseline in the development of the training and evaluation pipelines [29]. The work for this project is additionally open source to enable transparency and ease of future work. The code can be found on Github.