Information Bottlenecked Variational Autoencoder for Disentangled 3D Facial Expression Modelling

Hao Sun Nick Pears Yajie Gu
Department of Computer Science, University of York, UK
{hs1145, nick.pears, yg1390}@york.ac.uk

Abstract

Learning a disentangled representation is essential to build 3D face models that accurately capture identity and expression. We propose a novel variational autoencoder (VAE) framework to disentangle identity and expression from 3D input faces that have a wide variety of expressions. Specifically, we design a system that has two decoders: one for neutral-expression faces (i.e. identity-only faces) and one for the original (expressive) input faces respectively. Crucially, we have an additional mutual-information regulariser applied on the identity part to solve the issue of imbalanced information over the expressive input faces and the reconstructed neutral faces. Our evaluations on two public datasets (CoMA and BU-3DFE) show that this model achieves competitive results on the 3D face reconstruction task and state-of-the-art results on identity-expression disentanglement. We also show that by updating to a conditional VAE, we have a system that generates different levels of expressions from semantically meaningful variables.

1. Introduction

A 3D Morphable Model (3DMM) for human faces was proposed by Blanz and Vetter [3] more than 20 years ago. Since then, it has gained widespread use in a wide variety of both 2D and 3D applications. In more recent years, more non-linear 3D face models have being built that exploit powerful deep learning techniques. This has allowed more detailed reconstructions from more compressed latent representations [12]. Initially, models were built from neutral-expression faces only, but with more comprehensive datasets, newer approaches have also modelled facial expressions, for more general applications [14][25].

A key ability is to disentangle the identity part and the expression part from any input data (see Figure 1), and direct those disentangled parts into the corresponding model components. Such approaches can be beneficial for many applications, such as face reenactment and face recognition.

Figure 1. Disentangling identity from the full expressive face.

Here, we propose a concise architecture that improves disentanglement performance with fewer restrictions, compared to the state-of-the-art [40], and we evaluate the results, such as is given in Figure 1. To achieve this, we design two variational autoencoders (VAEs) for identity and expression separately, but are able to train them in an end-to-end manner without any pre-training. We employ the attention-based point cloud transformer (PCT) [17] as the encoder. This processes a set of points, which is unordered and without local neighborhood connectivity information. In other words, mesh topology is obviated, and we enable training on point cloud data for disentangled facial expression modelling. We also follow the idea of the information bottleneck in information theory, using an additional mutual information regulariser to encourage disentanglement and allow tuning of the compression of the latent representation. Furthermore, we utilise expression label information provided by the datasets by employing a *conditional* VAE as an upgrade to the proposed method. This enforces more disentangled expression information and thereby contributes to the explainability of the generative model.

In summary, our contributions are: 1) Incorporation of the point cloud transformer network, removing the need for a mesh vertex topology, and leveraging the high performance of attention-based architectures. 2) Use of an information bottleneck on the identity reconstruction subsystem to encourage improved identity and expression disentanglement. 3) Application of a conditional VAE as an upgrade to the proposed method to further disentangle expression information and generate expressions from semantically meaningful latent variables.

2. Related Work

2.1. 3D Facial Expression Modelling

Human face modelling focuses on building models of human faces to understand them. The 3D Morphable Model (3DMM), first proposed by Blanz and Vetter [3], is perhaps the most widely-employed technique in recent 3D face modelling applications. 3DMMs model a linear or non-linear 3D face space using a latent representation that may be constructed in various different ways. Examples include PCA [18, 4, 25], Gaussian mixture models [23], wavelet decomposition [6], dictionary learning [13] and neural nets [36]. The latent representation is sufficiently informative to reconstruct the original 3D faces to some level of accuracy. Here, we focus on neural net methods that are obtained by deep learning based 3D-to-3D autoencoders. A 3D-to-3D autoencoder based method means that it uses an encoder to extract the latent representation from the input 3D face, and a decoder to reconstruct the original input 3D face. Most current 3D face datasets use 3D meshes to represent their 3D face scans. A 3D mesh comprises a point cloud and a mesh topology. Depending on whether mesh topology information is utilised, encoder networks fall into two categories: (i) networks that process unordered point cloud data (*e.g*. PointNet [30], PCT [17]), and (ii) Graph Convolutional Networks (GCN) [5], which process sets of points with a predefined mesh topology (i.e. meshes). Recent GCN-based methods [2, 31, 24, 39] can only train on registered single datasets, however with PointNet, Liu *et al*. [26] train on combined multiple datasets with different topologies without point correspondence, which is a significant step towards reducing the limitations in the type of input data employed. We choose to employ an intermediate solution, such as [10], which uses registered point clouds only on single datasets and achieves better reconstruction and expression disentanglement results without topology information. Noting recent successes of transformer networks [37] in computer vision tasks [11], we use an open-sourced approach that applies this architecture to point cloud data (the Point Cloud Transformer (PCT) [17]) as our encoder.

3DMMs initially focused on modelling the variance over different identities of people, *e.g*. Basel Face Model 2009 (BFM09) [18], but latterly have added additional expression models to better model real faces that possibly appear with expressions. A number of works [4, 15, 31] model expression by modelling datasets that contain faces with expression, resulting in a set of identity-expression mixture coefficients. On the other hand, a number of models use separate coefficients for identity and expression [14, 25, 26, 7]. However, all aforementioned methods do not explicitly disentangle identity and expression and the two disentangling works that are most related to us, [19] and [40], both use the GCN (Graph Convolutional Network, [5]) as their encoder.

2.2. Disentangled 3D Facial Expression Modelling

To achieve facial expression disentanglement, the way in which identities and expressions are combined has to be defined. In the context of modelling identity and expression in two separate sets of coefficients, Egger *et al*. [12] classify the combination of identity model and expression model into three categories: additive, multiplicative and non-linear models. Zhang *et al*. [40] uses the additive assumption where expressions are represented in a blendshape that each vertex's coordinates can be directly added to the corresponded neutral face vertex's coordinates. Jiang *et al*. [19] use the non-linear model way that feeds both identity and expression latent code to a deep neural network and synthesis the final expression faces directly, which is the approach that we follow in our proposed method.

Jiang *et al*. [19] employed two networks, one removing identity and one removing expression from the input face, thus expecting the synthesised face to be the average face. They also synthesise the original face by a fusion network that combines the results from the identity remover and the expression remover. Zhang *et al*. [40] achieved the current state-of-the-art in 3D facial expression disentanglement results. They propose to add an objective that suggests independence between the identity latent code and the expression latent code by utilising a discriminator similar to the one that Kim and Mnih [20] proposed.

2.3. VAE Information-based Methods

There are a number of works that implement information-theoretic ideas into their VAE-based architecture; for example, the Variational Information Bottleneck (VIB) proposed by Alemi *et al*. [1]. Starting from the information bottleneck idea firstly proposed by Tishby *et al*. [34], Alemi *et al*. propose to optimise the information bottleneck using deep neural networks. This results in a similar autoencoder architecture to the VAE.

One of the main terms in information theory related to VAEs is the mutual information between the input and the latent code. However, analytically calculating this term requires a forward pass of the encoder network on the entire dataset for every backpropagation. This is undesired since it can increase the training time prohibitively. InfoGAN [9] uses Monte Carlo simulation to estimate a lower bound for the mutual information term directly. Kim *et al*. [20] and Zhang *et al*. [40] use the density ratio trick [28, 33] by introducing a discriminator. InfoVAE [41] obtains unbiased samples of the latent code by running a forward pass of decoder and encoder [16], and propose to use other divergences such as the Jensen-Shannon divergence. Here, we use Mini-batch Weighted Sampling (MWS), used in beta-TCVAE [8], to obtain a direct estimate of the aggregated prior without additional hyper-parameters or networks.

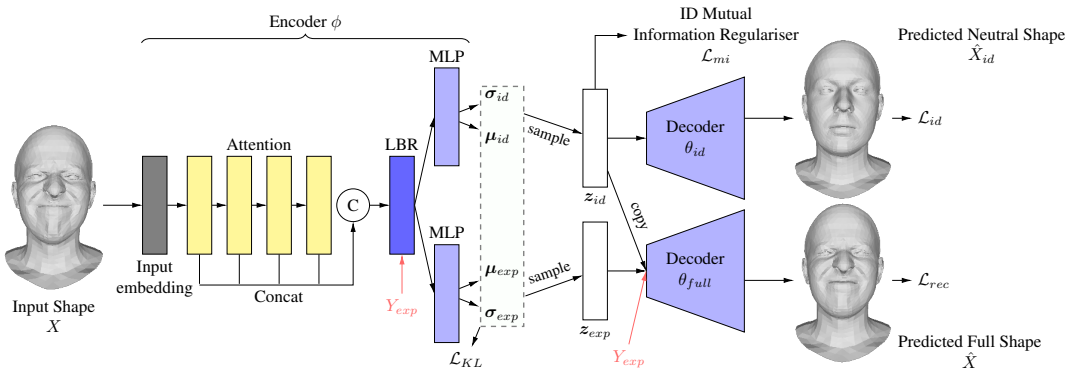

Figure 2. Overview of the architecture. *LBR* combines *Linear*, *BatchNorm* and *ReLU* layers. *MLP* stands for multi-layer perceptron. Additional conditional VAE add-ons are marked in red.

3. Proposed Method

3.1. Architecture

Denoting a 3D face point cloud $X_i \in \mathbb{R}^{M\times 3}$ where $i \in [1..N]$ and M is the number of points in each 3D face. We assume the dataset is comprised of $\left\{X_1, X_1^{id}, \ldots, X_N, X_N^{id}\right\}$, that means for each 3D face in the dataset, there is a corresponding identity face (i.e. neutral face). The goal is to reconstruct an identity face and full (expressive) face independently using their respective latent representations. We illustrate our architecture in Figure 2.

We separate the whole 3D facial expression modelling system into two sub-systems: identity (ID) VAE and full-face VAE, sharing the same encoder and trained simultaneously in an end-to-end manner. We follow the common VAE structure to build each sub-system in the first place. The point cloud transformer (PCT [17]) is used as our encoder network $q\left(\boldsymbol{z}_{id}, \boldsymbol{z}_{exp} \mid X, \phi\right)$ with learned weights ϕ that extracts features from 3D face point clouds, then predicts two sets of latent code: the identity latent code z_{id} and the expression latent code z_{exp}. We utilise two separate decoders $p\left(\hat{X}_{id} \mid \boldsymbol{z}_{id}, \theta_{id}\right)$ and $p\left(\hat{X} \mid \boldsymbol{z}_{id}, \boldsymbol{z}_{exp}, \theta_{full}\right)$ with weights θ_{id} and θ_{full} to reconstruct the identity face and the full face respectively. We cut off the gradient back-propagation flow for $\boldsymbol{z}_{id}$ from decoder θ_{full} to avoid updating the identity latent code with respect to errors that contain expression information.

Following the common VAE design for loss functions, we utilise reconstruction losses ($\mathcal{L}_{id}$ and $\mathcal{L}_{rec}$) and variational loss $\mathcal{L}_{KL}$ which will be explained in Section 3.2. In addition to the usual VAE structure, we utilise only an additional mutual information regularisation function on the identity latent code $\mathcal{L}_{mi}$, achieving significant improvement on disentanglement results compared to the current state of the art. We will explain the choice of this regulariser in Section 3.3, elaborate the two loss functions $\mathcal{L}_{id}$ and $\mathcal{L}_{mi}$ jointly as an information bottleneck. In Section 3.4, the four loss components are summed to give the loss function that enables end-to-end training of our network.

3.2. 3D Face VAE

3.2.1 Variational Autoencoder (VAE)

The VAE [22] is a generative model that aims to generate data based on approximated real data distribution $P\left(X\right)$ by conditioning real data on a latent vector, i.e. $P\left(X \mid \boldsymbol{z}\right)$. However, this often has an intractable posterior $P\left(\boldsymbol{z} \mid X\right)$. The VAE uses a deep neural network (denoted as decoder network) to approximate the conditional probability $P\left(X \mid \boldsymbol{z}\right)$ and uses another deep network (denoted as encoder network) to approximate decoder distribution's posterior distribution. To enable joint training of the two networks, the VAE uses a unified loss function named the Evidence Lower Bound (ELBO). This maximises the data distribution likelihood by maximising its lower bound. The ELBO loss function is formed as follows:

$$ELBO = -E_{q_\phi}\left[\log p_\theta\left(X \mid \boldsymbol{z}\right)\right] \quad (1)$$
$$+KL\left(q_\phi\left(\boldsymbol{z} \mid X\right) \parallel P\left(\boldsymbol{z}\right)\right), \quad (2)$$

where q_ϕ is the encoder, and p_θ the decoder.

3.2.2 Point Cloud Transformer (PCT)

We adopt the point cloud transformer (PCT, [17]) as our encoder to extract a latent code from input data. The PCT is an attention-based [37] network that processes unordered point sets and employs farthest point sampling and nearest neighbor search for input embedding. The core component, the attention module, takes the embedded point cloud inputs, and generates refined attention features based on global context by connecting all pairs of point clusters with attention weights. The attention feature is then fed into MLPs to generate identity and expression latent codes. Our architecture, using a PCT-based encoder, is depicted in Figure 2.

3.2.3 Our Proposed Method

To practically construct VAEs for both the identity subsystem and the full face sub-system, we have to build variational inference for the latent codes. Thus we let the encoder output the mean μ and the standard deviation σ of an isotropic Gaussian distribution $\mathcal{N}(\mu, diag(\sigma))$ that represents the latent code's distribution. Then the latent code is sampled from the predicted latent code distribution. Here we follow the original VAE paper [22] and use the reparameterisation trick [22] for differentiable sampling. Therefore, the KL loss (Equation 2) from the ELBO loss for both identity and expression latent code can be formed as:

$$\mathcal{L}_{KL} = KL\left(q_\phi\left(\boldsymbol{z}_{id}, \boldsymbol{z}_{exp} \mid X\right) \parallel \mathcal{N}(0, I)\right). \quad (3)$$

Meanwhile, it is a common practice to replace the reconstruction term 1 in the ELBO loss with a loss function that is used in non-variational deep learning tasks, such as the L1 norm used in [24, 19, 40] or Mean Squared Error (MSE). We adopt the MSE, and the two reconstruction losses for identity and full faces are:

$$\mathcal{L}_{id} = \|\hat{X}^{id} - X^{id}\|_2^2 \quad \text{and} \quad \mathcal{L}_{rec} = \|\hat{X} - X\|_2^2. \quad (4)$$

By doing so, we ensure the same reconstruction goal with the loss function that is practically proven to work well with stochastic gradient descent. This leaves the latent code to be the only variational part that is represented in distributions. The KL loss can then be seen as a regulariser that pulls them towards unit isotropic Gaussian distributions.

3.2.4 Additional Experiment: Conditional VAE

Several datasets provide corresponding labels along with their data. Utilising such information in a generative model can be beneficial for its performance and explainability. Sohn *et al*. [32] propose the Conditional Variational Autoencoder (CVAE), which utilises label information to allow the modelling of raw data conditioned on it. The main modification to the original VAE is to make both the encoder distribution q_ϕ and the decoder distribution p_θ condition on corresponding labels Y. The ELBO loss function in our setting for the full face VAE is then modified to:

$$\begin{aligned} ELBO_{cvae} = {} & E_z\left[\log P\left(X \mid \boldsymbol{z}_{id}, \boldsymbol{z}_{exp}, Y_{exp}\right)\right] \\ & - KL\left(q_\phi\left(\boldsymbol{z}_{exp} \mid X, Y_{exp}\right) \parallel P\left(\boldsymbol{z}_{exp} \mid Y_{exp}\right)\right), \end{aligned} \quad (5)$$

where Y_{exp} is a one-hot encoded expression label multiplied by the expression's level, which has the range $[0, 1]$.

As shown in Fig. 2 (red parts), we build a CVAE architecture upon our VAE architecture, by concatenating one-hot labels after the penultimate fully connected layer in our encoder. The one-hot encoded labels are also concatenated after the expression latent code, which is then passed to the full face decoder. With the CVAE, the trained decoder can generate new samples from given expression labels.

3.3. Mutual Information Regulariser

Due to the cost of 3D scans, most of the current 3D face datasets are obtained under specific experimental conditions rather than from in-the-wild. To increase the number of 3D faces collected, one has to acquire multiple scans of the same person. In this case, there exists groups of 3D face indices $K \subset [1..N]$, such that their corresponding 3D faces share the same corresponded neutral face, i.e. $X_k = X_{k'}, \forall k, k' \in K$. Therefore, the identity VAE part differs from the traditional VAE in two respects. Firstly, the ID decoder θ_{id} does not reconstruct the original input; rather, it reconstructs an expression-neutralised input, which has information content that is always less than or equal to that of the input. Secondly, assuming a single latent code, with the identity and the expression parts entangled, the ID decoder would have to reconstruct the same 3D face identity from different latent codes. That is, the identity latent code $\boldsymbol{z}_{id}$ would contain information about expressions on the input face. Thus, we propose an information bottleneck on the identity latent code, and address both challenges at the same time. We then modify it to work with a deep VAE model as a combination of a mutual information regulariser $\mathcal{L}_{mi}$ and the identity reconstruction loss $\mathcal{L}_{id}$. It simultaneously encourages the decoder to better reconstruct the identity face and forces the identity latent code to contain only the information of the reconstructed neutral faces and, therefore, achieves better identity and expression disentanglement.

3.3.1 Information Bottleneck

An information bottleneck is an information theory idea proposed by Tishby *et al*. [34]. The main idea is to build an objective function that jointly maximises the mutual information between the latent code and its reconstruction, and that minimises the mutual information between the input and the latent code. Putting more weight on the second of these terms allows for a more compressed latent representation [1].

In the identity sub-system scenario, the encoder encodes faces with expressions, which naturally introduces redundant expression information into the identity latent code, resulting in low compression efficiency. So we propose to put more weight on compression, jointly with reconstructing neutral faces, to eliminate expression information contained in the identity latent code. Also, the information bottleneck does not assume the reconstruction has to be identical to the input. Thus, to fit the information bottleneck to the identity sub-system, the objective can be formulated as:

$$J_{IB} = -I\left(X_{id}; \boldsymbol{z}_{id}\right) + I\left(\boldsymbol{z}_{id}; X\right), \quad (6)$$

Note that the Lagrangian multiplier in the second term from the original information bottleneck objective is ignored at

this stage, because the weight of the second term is built into the deep learning framework via hyper-parameters, described later in this section.

To begin with building this objective into the current 3D face VAE system, we reformulate the objective J_{IB}'s first term as:

$$I\left(X_{id}; \boldsymbol{z}_{id}\right) \approx \int_{X_{id}} \int_{\boldsymbol{z}_{id}} p\left(X_{id}, \boldsymbol{z}_{id}\right) \log p\left(X_{id} \mid \boldsymbol{z}_{id}\right) d\boldsymbol{z}_{id} dX_{id} \quad (7)$$

Following the assumption described in the architecture, we have:

$$p\left(X_{id}, \boldsymbol{z}_{id}\right) = \int_{X} p\left(\boldsymbol{z}_{id} \mid X\right) p\left(X, X_{id}\right) dX. \quad (8)$$

Since X and X_{id} form a data point in the dataset of size N, we can estimate $p\left(X, X_{id}\right)$ using the dataset (i.e. empirical data distribution), then further derive an empirical lower bound of the mutual information in Equation 7:

$$I\left(X_{id}; \boldsymbol{z}_{id}\right) \geq \frac{1}{N} \sum_{n}^{N} E_{\boldsymbol{z}_{id} \sim q_{\phi}}\left[\log p_{\theta_{id}}\left(X_{n}^{id} \mid \boldsymbol{z}_{id}\right)\right], \quad (9)$$

where the encoder network q_{ϕ} is used to estimate the conditional probability $p\left(\boldsymbol{z}_{id} \mid X_{n}\right)$ and the identity decoder network $p_{\theta_{id}}$ is used to estimate the conditional probability $p\left(X_{n}^{id} \mid \boldsymbol{z}_{id}\right)$, thus we have the final result in Equation 9. By comparing with the reconstruction loss in the original ELBO loss in Equation 1, we note that Equation 9 is an aggregated negative likelihood of the reconstructed identity faces. In order to take advantage of mini-batch training and stochastic gradient descent, we use mean squared error loss $\mathcal{L}_{id}$ in Equation 4 to replace the original variational reconstruction loss. Using stochastic gradient descent to backpropagate from the loss function $\mathcal{L}_{id}$ on a mini-batch basis can be seen as a practically effective way of estimating the gradient of the aggregated negative likelihood in Equation 9 over the whole dataset. Thus we encourage better reconstruction of the identity face, by effectively maximising the mutual information between X_{id} and $\boldsymbol{z}_{id}$.

Using the encoder network to estimate the conditional probability $p\left(\boldsymbol{z}_{id} \mid X\right)$ in second mutual information term in Equation 6, results in the mutual information loss $\mathcal{L}_{mi}$, given as:

$$\mathcal{L}_{mi} = I_{q}\left(\boldsymbol{z}_{id}; X\right) = E_{X}\left[KL\left(q_{\phi}\left(\boldsymbol{z}_{id} \mid X\right) \parallel q_{\phi}\left(\boldsymbol{z}_{id}\right)\right)\right]. \quad (10)$$

However, obtaining the aggregated posterior $q_{\phi}\left(\boldsymbol{z}_{id}\right) = E_{X}\left[q_{\phi}\left(\boldsymbol{z}_{id} \mid X\right)\right]$ directly can be undesirable, since it requires a forward pass of the entire dataset on the encoder network for each backpropagation [20]. Therefore we applied the mini-batch weighted sampling (MWS) technique proposed by [8], which was inspired by importance sampling, to estimate $q_{\phi}\left(\boldsymbol{z}_{id}\right)$. Suppose we have a mini-batch of $\{X_1, \ldots, X_B\}$, the estimator is formed as:

$$E_{q_{\phi}\left(\boldsymbol{z}_{id}\right)}\left[q_{\phi}\left(\boldsymbol{z}_{id}\right)\right] \approx \frac{1}{B} \sum_{i}^{B} \left[\log \frac{1}{NB} \sum_{j}^{B} q_{\phi}\left(\boldsymbol{z}_{id}\left(X_i\right) \mid X_j\right)\right], \quad (11)$$

where $\boldsymbol{z}_{id}\left(X_i\right)$ is a sample from $q_{\phi}\left(\boldsymbol{z}_{id} \mid X_i\right)$.

We have formed an information bottleneck on identity faces, resulting in the combination of two loss functions $\mathcal{L}_{id}$ and $\mathcal{L}_{mi}$. However, the original information bottleneck introduces a Lagrangian multiplier to allow tuning of the compression level. Since we replaced the variational reconstruction loss with a MSE loss, both loss functions have to be re-weighted to correctly balance the training process. Thus we introduce two hyper-parameters β_{id} and β_{mi} for this purpose. To strengthen the information bottleneck, one can increase β_{id} for better reconstructed identity faces and increase β_{mi} for a more compressed identity latent code.

3.4. Final Loss Function

To balance the reconstruction loss and KL loss, two additional hyper-parameters are introduced, resulting in the full loss function:

$$\mathcal{L} = \lambda_1\left(\mathcal{L}_{rec} + \beta_{id}\mathcal{L}_{id}\right) + \lambda_2\left(\mathcal{L}_{KL} + \beta_{mi}\mathcal{L}_{mi}\right). \quad (12)$$

Where we divide four loss components into two groups, balancing them with λ_1 and λ_2, then increasing the information bottleneck weights β_1 and β_2 to strengthen its effect.

3.5. Implementation Details

PyTorch [29] is used to build the whole system. The encoder PCT uses the original PCT paper's architecture on the self-attention module, followed by fully connected layers that are configured as $\{1024/256/64/\left(\|\boldsymbol{z}_{id}\| + \|\boldsymbol{z}_{exp}\|\right) \times 2\}$. For decoders, the identity decoder and the full face decoder share the same architecture: an MLP with 256 hidden neurons. For a fair comparison with other models the evaluate on the CoMA dataset, we choose latent code sizes as $\|\boldsymbol{z}_{id}\| = 4$ and $\|\boldsymbol{z}_{exp}\| = 4$. For the loss function weights, we use $\lambda_1 = 6.6 \times 10^{-2}$, $\lambda_2 = 3 \times 10^{-3}$, $\beta_{id} = 10$, $\beta_{mi} = 50$. The whole system is trained over 300 epochs with the Adam [21] optimiser and we set the learning rate to 5×10^{-5} with a L2 weight decay [27] set to 10^{-4} and a learning rate decay of 0.7 for every 50 epochs. The KL loss and mutual information regulariser weight λ_2 decays linearly to 0 over 350 epochs.

4. Evaluation

We now evaluate the performance of our proposed system. After presenting the two datasets and the evaluation metrics, we compare our proposed VAE with three state-of-the-art systems in quantitative evaluations. We then present an ablation study and, finally, qualitative results for both proposed VAE (Figure 4 and Figure 5) and conditional VAE (Figure 6 and Figure 7) are presented. Note that all experiments are based on point clouds only, with the mesh topology only used for visualisation.

4.1. Datasets

CoMA Dataset [31] This contains scans of 12 individuals performing 12 different expressions. For each subject-expression pairing, there is a video of that person making the desired expression, giving a total of $20,466$ 3D scans in the dataset. All of the 3D face scans are registered with FLAME topology [25] and are pose normalised. We follow the data split scheme proposed by [19] and [31] that sorts all videos in alphabetical order, and then takes 10 frames for every 100 frames as the test set and train on the reminder.

BU-3DFE [38] This contains 100 individuals each with 6 different expressions over 4 different expression levels. For each subject, one neutral scan is performed, resulting in a total of $2,500$ scans. All the 3D face are registered to the same topology. In order to further normalise the pose, we perform a rigid registration of all 100 neutral faces to their mean based on a number of landmarks. Then for each subject, their 24 expression scans are rigidly registered with the neutral face based on a number of expression invariant key points. Finally, following [40], the first 10 subjects are selected as the test set and the rest are used for training.

4.2. Evaluation Metrics

We employ the same evaluation metrics as the most closely related papers [19, 40], namely reconstruction error and disentanglement error (this is exactly the same error as the decomposition error used in [40]).

Reconstruction Error The fundamental metric for a generative model is reconstruction error. Since all the vertices are corresponded to the ground truth, we can use the Average Vertex (Euclidean) Distance (AVD) to measure the reconstruction quality:

$$E_{rec} = \frac{1}{M}\sum_{j}^{M}\left\|\hat{X}_j - X_j\right\|_2, \tag{13}$$

where M is the number of vertices in a single face.

Disentanglement Error The disentanglement error measures the variance in the predicted identity faces from the same subject. Given a subset of the test set that contains various expressions from the subject d denoted as: $\{\mathcal{M}_i\}$, the predicted identity faces (i.e. neutral faces) can be generated by the system, denoted as: $\left\{\mathcal{M}_i^{id}\right\}$. Let $\mathcal{M}^d$ denote the mean face of all predicted neutral faces for subject d. The disentanglement error can then be formulated by:

$$E_{dis} = \sigma\left(\left\{\left\|\mathcal{M}_{ij}^{id} - \mathcal{M}_j^d\right\|_2\right\}\right), \tag{14}$$

where j is the vertex index and σ is the standard deviation function. Jiang *et al*. [19] propose to apply the same error metric on predicted expressions from different subjects that perform the same expression. We omit this analysis, because we assume different subjects perform the same expression in different ways, thus the expression disentanglement error is expected to be high. However, we use a *conditional* VAE to further disentangle expression information.

4.3. VAE Quantitative Evaluation

Compared Methods We compare our work to a number of 3D face modelling methods based on the autoencoder structure. MeshAE [31] and SpiralNet++ [15] both focus on applying a GCN architecture to 3D-to-3D mesh reconstruction, regardless of disentanglement. FLAME [25] builds identity and expression latent representations separately and reconstructs using a linear system. The two most related works to our proposed system are Jiang *et al*. [19] and Zhang *et al*. [40]. Both of these focus on disentangled facial expression modelling using GCN architectures and are evaluated using same metrics.

Table 1 gives disentanglement results for several systems. Our baseline, denoted as "Ours - No IB" which stands for no *Information Bottleneck*, sets β_{id} to 1 and β_{mi} to 0 and obtains a competitive result. One intermediate result "Ours - $\beta_{mi} = 0$" sets $\beta_{id} = 10$ and $\beta_{mi} = 0$ shows the effectiveness of the $\mathcal{L}_{mi}$ solely. Our final proposed method sets $\beta_{id} = 10$ and $\beta_{mi} = 50$ and surpasses the current state-of-the-art by a large margin.

Method	mean	median
FLAME[25]	0.599	0.591
Jiang *et al*. [19]	0.064	0.062
Zhang *et al*. [40]	0.019	0.020
Ours - No IB	0.025	0.022
Ours - $\beta_{mi} = 0$	0.016	0.013
Ours	**0.006**	**0.005**

Table 1. Disentanglement result (mm) compared with current literature on CoMA dataset.

The detailed results of reconstruction error are shown in Table 2. The reconstruction results are divided into two groups, where non-disentangling methods generally have better reconstruction results. One potential reason is that

disentanglement methods add extra objectives for disentanglement that act similar to regularisers, which will make reconstruction performance drop. However, from the results, our model can still achieve competitive reconstruction results. Furthermore, by comparing the two results of our model, the performance drop on reconstruction quality introduced by the mutual information regulariser is a tolerable price to pay for disentanglement.

	Method	mean ± std	median	ne mean ± std	median
Non-disentanglement Methods	MeshAE[31]	0.845 ± 0.994	0.496	\	\
	SpiralNet++[15]	0.543 ± 0.663	0.320	\	\
	Ours - No IB	0.614 ± 0.192	0.594	0.065 ± 0.021	0.065
	Ours - $\beta_{mi} = 0$	0.604 ± 0.183	0.581	0.054 ± 0.020	0.049
Disentanglement Methods	FLAME[25]	1.451 ± 1.649	0.871	\	\
	Jiang *et al.* [19]	1.413 ± 1.639	1.017	\	\
	Zhang *et al.* [40]	0.665 ± 0.748	**0.434**	\	\
	Ours	$\mathbf{0.663 \pm 0.215}$	0.643	$\mathbf{0.051 \pm 0.021}$	0.048

Table 2. Reconstruction results: Average Vertex Distance (mm) compared with literature on the CoMA dataset (column 3 and 4). Our methods' reconstructed neutral faces AVD (column 5 and 6).

The results for disentanglement error and reconstruction error on the BU-3DFE dataset are shown in Table 3. Our approach here employs $\beta_{id} = 1$ and $\beta_{mi} = 50$ to obtain a better disentanglement result compared to current state-of-the-art, again with a competitive reconstruction error.

Method	Disentanglement Error		Reconstruction Error	
	mean	median	mean ± std	median
FLAME[25]	0.600	0.632	2.596 ± 2.055	2.055
Jiang *et al.* [19]	0.611	0.590	2.054 ± 1.199	1.814
Zhang *et al.* [40]	0.361	0.327	$\mathbf{1.551 \pm 0.924}$	**1.375**
Ours	**0.328**	**0.296**	1.628 ± 0.333	1.589

Table 3. Disentanglement results (mm) compared with current literature on the BU-3DFE dataset.

4.4. Ablation Studies

The effect of introducing the information bottleneck is now evaluated. In Figure 3a and Figure 3b, we study the impact of modifying the mutual information regulariser's weight $\beta_{mi} \in \{0, 10, 25, 50, 75, 100\}$ while keep the identity reconstruction loss weight at fixed values $\beta_{id} \in \{0, 10\}$. From the graphs, one can observe that increasing identity loss weight and mutual information regulariser weight can result in lower disentanglement error and higher reconstruction error. The increase in reconstruction error is expected because there exists a fundamental trade-off for the information bottleneck (IB) between concise representation and good reconstruction power [35]. The mutual information regulariser encourages the first term, while the reconstruction errors encourage the other. Meanwhile, using overly large IB weights can harm performance. From the graphs, one can observe that given an identity reconstruction loss is not strongly weighted, the information bottleneck can constrain the necessary information to convey from input to latent code, resulting in an overly compressed latent representation. Therefore, it is critical to adjust the information

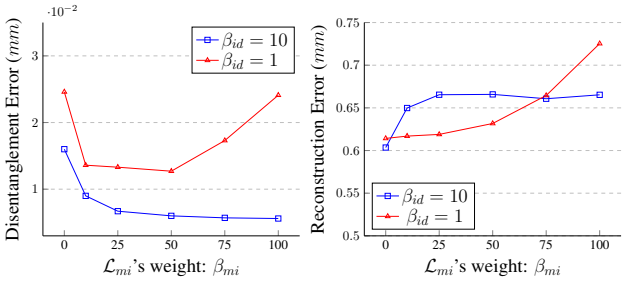

Figure 3. Mutual information regulariser weights' impact on: (a) disentanglement error and (b) reconstruction error, CoMA dataset

bottleneck to the appropriate level, which can raise the difficulty in hyper-parameter tuning in practice. Also, another drawback of the system is that the disentanglement error can have a relatively larger variance. When repetitively training three times without and with the mutual information regulariser, the variance of disentanglement error raises from 0.0002 to 0.0014. This is because we use sampling to obtain the mutual information term. Finally, we apply the mesh topology to the reconstructed point clouds to evaluate mesh quality in regard to self-intersecting faces (fewer is better). On the CoMA dataset, compared to the ground truth data, which has 548.60 intersecting faces per mesh, the reconstructed data has an average of 510.06 intersecting faces per mesh. For the BU3DFE dataset, the number of intersecting faces per-mesh is 0.012 and 0.020 for ground truth and reconstruction respectively.

Although we obtained the state-of-the-art disentanglement result on BU-3DFE dataset, the performance on unseen identities in the BU-3DFE dataset remains challenging, as the information bottleneck is not as effective as when it is applied to the CoMA dataset. We evaluate the effectiveness of mutual information loss on BU-3DFE. Raising β_{mi} from 0 to 50 only results in a small decrease of the disentanglement error from 0.332 to 0.328. The reconstructed neutral for $\beta_{mi} = 50$ has an average vertices distance of $2.264\,\mathrm{mm}$, increased from $2.253\,\mathrm{mm}$ when $\beta_{mi} = 0$. The main reasons for the non-ideal overall performance on BU-3DFE are: 1) part of the faces are not well registered with each other, resulting in small pose differences and noise; 2) lack of data causes difficulties in avoiding overfitting in the current setup, and $\mathcal{L}_{mi}$ on a smaller dataset is not as effective as on a larger dataset *e.g.* CoMA; 3) each subject contains only 24 faces with expressions, which is insufficient compared to $\sim 1,200$ per subject in the CoMA dataset; 4) 100 different identities are present, making it easier to learn the variation of expressions than identities, thus the neutral AVD is higher.

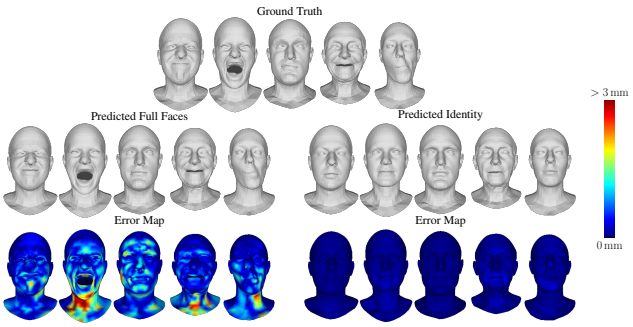

Figure 4. Visualisation of reconstruction quality on both identity and full face.

4.5. VAE Qualitative Evaluation

In Figure 4, we demonstrate the reconstruction quality for both identities and full faces (i.e. with expression). With the information bottleneck applied, the reconstructed neutral faces have extremely low error.

In Figure 5, we visualise the identity latent code on the model trained on BU-3DFE dataset by dimension reduction using PCA. We show that the learned model clusters the latent representation for similar faces.

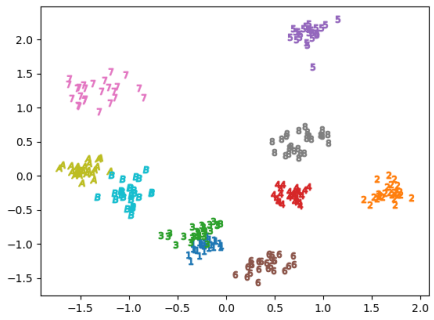

Figure 5. Scatter plot of PCA-processed identity latent code on BU-3DFE dataset. The axes represent dimensionally-reduced latent code values. Different subjects are marked with their hexadecimal ID and with different colours.

4.6. Conditional VAE Evaluation

We also perform evaluations using a CVAE using the same hyper-parameters as the proposed method on CoMA data. By providing 12 explainable expression-level variables to the decoder, 3 out of 4 expression latent variables are collapsed to the standard Gaussian distribution, while the remaining variable encodes mouth direction (visual details in Figure 6). This reduces the uninterpretable latent variables from 8 to 5. Our CVAE gives a reconstruction error of $0.740\,\mathrm{mm}$ and disentanglement error of $0.008\,\mathrm{mm}$. It enables the generation of expressions with semantic expression labels in exchange for a small performance drop.

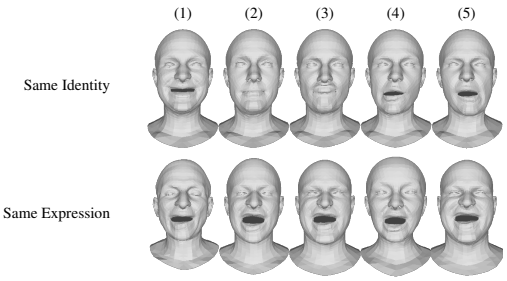

Figure 6. Conditional VAE generated samples. Upper row: same identity, different expressions; Lower row: same expression (mouth extreme), different identities. The expressions generated on the first row are: (1) bare teeth. (2) eyebrow, (3) lips up, (4) mouth side (left) and (5) mouth side (right)

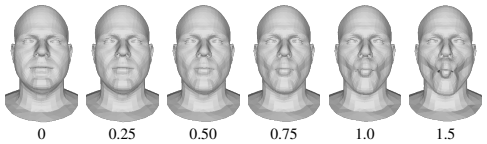

Figure 7. Conditional VAE generated samples with gradual increasing expression level on *cheeks in* expression.

In Figure 6, we use the full face decoder part of the conditional VAE to generate faces directly. The upper row demonstrates the different expressions generated upon a fixed identity by providing an expression level for selected expression. We can also control which expression to generate by changing the corresponding variables. The CoMA dataset only provides one label for the mouth going to both left and right side, however, the conditional VAE still captures that information and stores it as one variable in $\boldsymbol{z}_{exp}$. Given other latent variables are collapsed to a prior, modifying this uninterpretable variable results in (4) and (5) on the first row of Figure 6. The bottom row shows generating the *mouth extreme* expression using different identities. Finally, Figure 7 shows that with the CVAE, one can generate certain expressions with different levels of intensity.

5. Conclusion

We demonstrated identity and expression disentanglement, using an intuitive structure with an additional information bottleneck on the identity sub-system. We showed that the information bottleneck can be integrated with the current VAE training structure by adding an additional mutual information loss. Future work may include finding the optimal boundary for optimised weight selection and further increasing the method's efficiency for datasets with fewer scans per subject. Our results show that the our architecture performs better than the current state-of-the-art in term of disentanglement performance. Furthermore, with use of a CVAE, we are able to generate expressions using expression labels and their corresponding expression levels.

References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[2] Timur Bagautdinov, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. Modeling facial geometry using compositional vaes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2018.

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

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

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[6] Alan Brunton, Timo Bolkart, and Stefanie Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

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

[8] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in vaes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2615–2625, 2018.

[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

[10] Hang Dai and Ling Shao. Pointae: Point auto-encoder for 3d statistical shape and texture modelling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5410–5419, 2019.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

[13] Claudio Ferrari, Giuseppe Lisanti, Stefano Berretti, and Alberto Del Bimbo. Dictionary learning based 3d morphable model construction for face recognition with varying expression and pose. In *2015 International Conference on 3D Vision*, pages 509–517. IEEE, 2015.

[14] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82. IEEE, 2018.

[15] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[16] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[17] Menghao Guo, Junxiong Cai, Zhengning Liu, Taijiang Mu, Ralph R Martin, and Shimin Hu. Pct: Point cloud transformer. *arXiv preprint arXiv:2012.09688*, 2020.

[18] IEEE. *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, Genova, Italy, 2009.

[19] Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. Disentangled representation learning for 3d face shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11957–11966, 2019.

[20] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

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

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

[23] Paul Koppen, Zhen-Hua Feng, Josef Kittler, Muhammad Awais, William Christmas, Xiao-Jun Wu, and He-Feng Yin. Gaussian mixture 3d morphable face model. *Pattern Recognition*, 74:617–628, 2018.

[24] Kun Li, Jingying Liu, Yu-Kun Lai, and Jingyu Yang. Generating 3d faces using multi-column graph convolutional networks. In *Computer Graphics Forum*, volume 38, pages 215–224. Wiley Online Library, 2019.

[25] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017.

[26] Feng Liu, Luan Tran, and Xiaoming Liu. 3d face modeling from diverse raw scan data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9408–9418, 2019.

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[28] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison,

Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

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

[32] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

[33] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012.

[34] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[35] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[36] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355, 2018.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[38] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE, 2006.

[39] Cunkuan Yuan, Kun Li, Yu-Kun Lai, Yebin Liu, and Jingyu Yang. 3d face reprentation and reconstruction with multi-scale graph convolutional autoencoders. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1558–1563. IEEE, 2019.

[40] Zihui Zhang, Cuican Yu, Huibin Li, Jian Sun, and Feng Liu. Learning distribution independent latent representation for 3d face disentanglement. In *2020 International Conference on 3D Vision (3DV)*, pages 848–857. IEEE, 2020.

[41] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.