3D Shape Variational Autoencoder Latent Disentanglement via Mini-Batch Feature Swapping for Bodies and Faces

Simone Foti Bongjin Koo Danail Stoyanov Matthew J. Clarkson
University College London
s.foti@cs.ucl.ac.uk

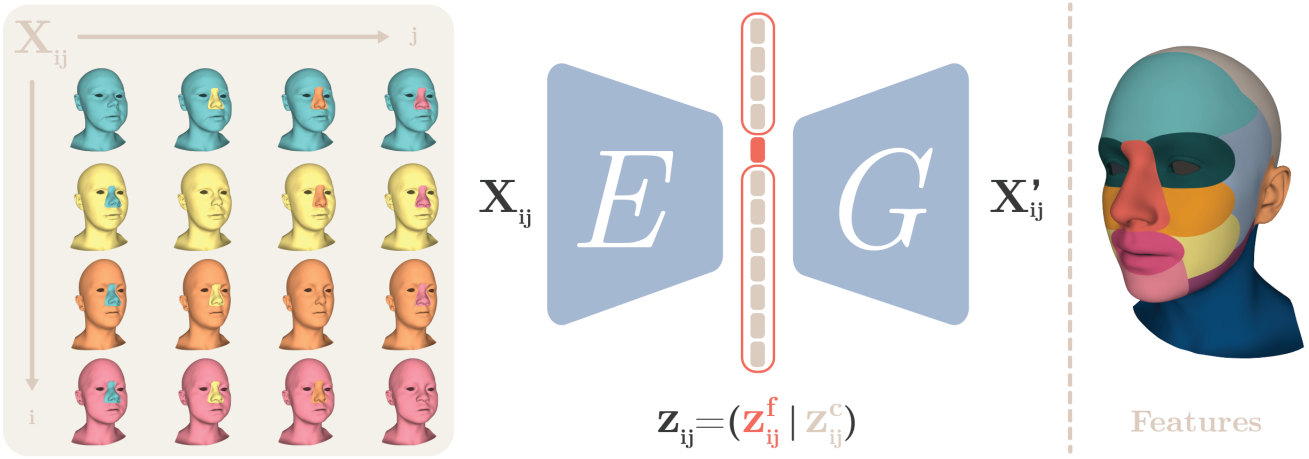

Figure 1. Schematic Description of the Proposed Method. *Left*: an arbitrary identity feature is selected and a mini-batch of vertices ($\mathbf{X}_{ij}$) is created by swapping features across different 3D shapes. Colours represent identities. Notice that features from the same identity have the same colour. *Centre*: a 3D-VAE ($\{E, G\}$) encodes $\mathbf{X}_{ij}$ in its latent representations $\mathbf{z}_{ij} = (\mathbf{z}^f_{ij}|\mathbf{z}^c_{ij})$, which are subsequently decoded into $\mathbf{X}'_{ij}$. In this case f corresponds to the nose. Therefore, while $\mathbf{z}^f_{ij}$ controls the shape of the nose, $\mathbf{z}^c_{ij}$ controls the shape of the rest of the face. *Right*: visual representation of all the different mesh features for which we seek to obtain a disentangled latent representation.

Abstract

Learning a disentangled, interpretable, and structured latent representation in 3D generative models of faces and bodies is still an open problem. The problem is particularly acute when control over identity features is required. In this paper, we propose an intuitive yet effective self-supervised approach to train a 3D shape variational autoencoder (VAE) which encourages a disentangled latent representation of identity features. Curating the mini-batch generation by swapping arbitrary features across different shapes allows to define a loss function leveraging known differences and similarities in the latent representations. Experimental results conducted on 3D meshes show that state-of-the-art methods for latent disentanglement are not able to disentangle identity features of faces and bodies. Our proposed method properly decouples the generation of such features while maintaining good representation and reconstruction capabilities. Our code and pre-trained models are available at github.com/simofoti/3DVAE-SwapDisentangled.

1. Introduction

The generation of 3D human faces and bodies is a complex task with multiple potential applications ranging from movie and game productions, to augmented and virtual reality, as well as healthcare applications. Currently, the generation procedure is either manually performed by highly skilled artists or it involves semi-automated avatar design tools. Even though these tools greatly simplify the design process, they are usually limited in flexibility because of the intrinsic constraints of the underlying generative mod-

els [17]. Blendshapes [28, 31, 38], 3D morphable models [5,25,33], autoencoders [3,8,16,35], and generative adversarial networks [1,7,15,24] are currently the most used generative models, but they all share one particular issue: the creation of local features is difficult or even impossible. In fact, not only do generative coefficients (or latent variables) lack any semantic meaning, but they also create global changes in the output shape. For this reason, we focus on the problem of 3D shape creation by enforcing disentanglement among sets of generative coefficients controlling the identity of a character.

Following [4,18,19] we define a disentangled latent representation as one where changes in one latent unit affects only one factor of variation while being invariant to changes in other factors. More interpretable and structured latent representations of data that expose their semantic meaning have been widely researched in the artificial intelligence community [10,13,18,19,22], but this is still an open problem especially for generative models of 3D shapes [3]. Given the superior representation capabilities, the reduced number of parameters, and the stable training procedures, we decide to focus our study on deep-learning-based generative models and in particular on variational autoencoders (VAEs). In this field, recent work has tried to address the latent disentanglement problem for 3D shapes and managed to decouple the control over identity and expression (or pose) [1,3,8], but they are still unable to properly disentangle identity features. Some success has been achieved in the generation of 3D shapes of furniture [29, 43], but the structural variability of the data requires complex architectures with multiple encoders and decoders for different furniture parts. In contrast, our method relies on a single VAE which is trained by curating the mini-batch generation procedure and with an additional loss. The intuition behind our method is that if we swap features (e.g. nose, ears, legs, arms, etc.) across the input data in a controlled manner (Fig. 1, *Left*), we not only know a priori which shapes within a mini-batch have (do not have) the same feature, but we also know which are (are not) created from the same face (body). These differences and similarities across shapes should be captured in the latent representation. Therefore, assuming that different subsets of latent variables correspond to different features, we can partition the latent space and leverage the structure of the input batch to encourage a more disentangled, interpretable, and structured representation.

With the objective of building a model capable of generating 3D meshes, we define our VAE architecture extending [16]. This state-of-the-art model proved to be fast and capable of better capturing non-linear representations of 3D meshes, while leveraging very intuitive convolutional operators characterised by a reduced number of parameters. Nonetheless, the network choice is arbitrary and we expect our method to be working also with other network configurations and operators. Even though we consider meshes as our primary data structures, it is also worth noting that, by providing semantic segmentations of the different features, our method is applicable to voxel- or point-cloud-based generative models. We believe that the generality of the proposed method is particularly important in the current geometric deep learning field, where definitions of 3D convolutions and pooling operators are still an open problem.

To summarise, the key contributions of our approach are: (*i*) the definition of a new mini-batching procedure based on feature swapping, (*ii*) the introduction of a novel loss function capable of leveraging shape differences and similarities within each mini-batch, and (*iii*) the consequent creation of a 3D-VAE capable of generating 3D meshes from a more interpretable and structured latent representation.

2. Related Work

In this section, we first discuss existing work on 3D generative models of faces and bodies, followed by state-of-the art approaches for latent disentanglement of autoencoder-based generative models.

Generative Models Blendshape models manually created by artists linearly interpolate local features between two or more manually selected shapes. These models are common as consumer-level avatar design tools adopted by several videogame engines. Even though they guarantee control over the generation of local features, they are very large models usually built with only a few subjects and are capable of offering only very limited flexibility and expressivity [17]. A widespread approach to overcome these limitations is to rely on linear statistical 3D morphable models (3DMM). These models are based upon the identity space of a population of 3D shapes, and are usually built by applying a principal component analysis (PCA) over the entire dataset. They are always built with the assumption that shapes are registered between each other and in dense point correspondence. This allows the generation of meaningful and morphologically realistic shapes as linear combinations of training data. This technique was pioneered by [5] and further developed and adopted by many researchers [12]. Interestingly, [17] divided the face in different local patches and trained a PCA model for each region in order to control the generation of different facial features. The generation of new faces and interactive face sculpting are then achieved through a constrained optimisation. Recently, [32,33] combined multiple 3DMMs to create the first combined, large-scale, full-head morphable model. In particular, the universal head model (UHM) [33] combines the Large-Scale Face Model (LSFM) [6], which was built with face scans from $10,000$ subjects, with the LYHM head model [9]. In [32] it was extended by combining also a detailed ear model, eye

and eye region models, as well as basic models for mouth, teeth, tongue and inner mouth cavity. As further detailed in Sec. 4, given the high diversity of UHM we decided to train our face model on heads from [33].

PCA-based models and blendshapes are often combined. For instance, SMPL [28] learns linear PCA models of male and female body shapes from approximately $2,000$ scans per gender, and subsequently uses the resulting principal components as body shape blendshapes capable of efficiently controlling the identity of a subject. The same approach is used also by STAR [31], which not only creates more realistic pose deformations than [28], but it also leverages $10,000$ additional scans to improve the generalisation capabilities of the model. Given its better generalisation with respect to other state-of-the-art methods, we trained our body model on shapes generated from STAR.

Recently, advances in the geometric deep learning community allowed to efficiently define convolutional operators on 3D data such as meshes and point-clouds. [35] is the first AE for 3D meshes of faces based on a graph convolutional neural network. This model was built using significantly less parameters than state-of-the-art PCA-based models and showed lower reconstruction errors as well as better generalisation to unseen faces. Other AE-based architectures leveraging different convolutional operators over different datasets were subsequently introduced [3, 8, 26, 44, 46]. Despite the remarkable performance of these models, we decided to adopt the base architecture of [16], which further improved upon previous methods by defining a more intuitive convolutional operator based on dilated spiral convolutions (i.e. spiral++ convolution).

An alternative line of work considers generative adversarial networks (GANs) instead of autoencoders. The first GAN operating on 3D meshes was proposed in [7] and it allowed to disentangle identity from expression generative factors. Other methods usually map 3D shapes to the image domain and then train adversarial networks with traditional 2D convolutions [1, 15, 24]. GAN models are generally able to generate more detailed and realistic 3D shapes than autoencoders at the cost of being more unstable and difficult to train.

As aforementioned in Sec. 1, with the exception of artistically-created blendshape models and [17], none of the other methods here described allow to control local changes during the generation process because their generative coefficients lack any semantic meaning, are not easily interpretable and are not properly disentangled.

Autoencoder Latent Disentanglement Latent disentanglement for the generation of 3D shapes has been explored mostly in relation to the disentanglement of identity and pose generative factors. [3] created a two level architecture combining a point-cloud AE with a VAE where the latent space is successfully partitioned by relying on multiple geometric losses and disentanglement penalties. [8] achieves similar results by training a point-cloud VAE while controlling the amount of distortion incurring in the construction of the latent space. As mentioned in Sec. 1, these methods are not capable of disentangling generative factors controlling the identity of different subjects. Methods such as [29, 43], on the other hand, are able to control different parts of furniture meshes, but they require complex architectures with multiple encoders and decoders controlling the different parts. Even though part hierarchies have to be considered in the model formulation, differently from faces and bodies, discontinuities between different parts are not a problem when generating furniture.

Research on latent disentanglement of AEs often focuses on the scenario in which only raw observations are available without any supervision about the generative factors, and it is usually performed on images. [18] proposed a simple modification to a VAE [21]. By increasing the weight of the Kullback–Leibler (KL) divergence, β-VAE showed better latent disentanglement properties at the expense of a reduced quality of the generated samples. Subsequent work, such as [19, 23], tried to overcome this limitation. The DIP-VAE [23] leverages an additional regularisation term on the expectation of the approximate posterior over observed data. The Factor VAE [19] encourages the latent distribution to be factorial, and therefore independent across dimensions, by using a latent discriminator and by adding a total correlation term in the VAE loss function. An interesting approach to encourage latent variables to represent predefined transformations was proposed in [22], where mini-batches are created combining active and inactive transformations and gradients influencing the latent are modified during backpropagation. However, this method requires synthetic datasets created with known properties that can be used during training to achieve the disentanglement. Recently, [13] proposed a VAE in which the objective function is hierarchically decomposed to control the relative levels of statistical independence between groups of variables and for individual variables in the same group. The recursive formulation of the loss introduces additional terms for any variable that has to be disentangled and works only where the factors of variation are uncorrelated scalar variables, a requirement that hampers the applicability of the model in real-world scenarios. Finally, the Guided-VAE [10] in its unsupervised setting leverages a secondary decoder that learns a set of PCA bases that are used to guide the training over simple geometrical shapes. Nevertheless, being the secondary decoder based on a PCA, latent variables suffer the same problems of PCA models.

Among the aforementioned methods for latent disentanglement the DIP-VAE [23] and Factor VAE [19] showed good disentanglement performance also on in-the-wild im-

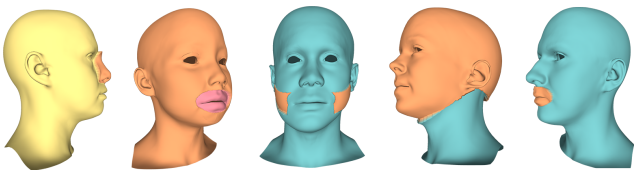

Figure 2. Examples of feature swapping for different features and different subjects.

age datasets while requiring only minor modifications to the VAE formulation. For this reason, we implemented a DIP-VAE and a Factor VAE operating on meshes and compared them against our method.

3. Method

The proposed method (Fig. 1) allows us to obtain more interpretable and structured latent representations for self-supervised 3D generative models. This is achieved by training a mesh-convolutional variational autoencoder (Sec. 3.1) with a mini-batch controlled feature swapping procedure and a latent consistency loss (Sec. 3.2).

3.1. Mesh Variational Autoencoder

A manifold triangle mesh is defined as $\mathcal{M} = \{\mathbf{X}, \boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{F}}\}$, where $\mathbf{X} \in \mathbb{R}^{N\times 3}$ is its vertex embedding, $\boldsymbol{\mathcal{E}} \in \mathbb{N}^{\varepsilon\times 2}$ is the edge connectivity that defines its topology, and $\boldsymbol{\mathcal{F}} \in \mathbb{N}^{\Gamma\times 3}$ are its triangular faces. Assuming that meshes share the same topology across the entire dataset, $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{F}}$ are constant and meshes differ from one another only for the position of their vertices, which are assumed to be consistently aligned, scaled, and with point-wise correspondences. Since traditional convolutional operators are not compatible with the non-Euclidean nature of meshes, we build our generative model with the simple yet efficient approach defined in [16]. Convolution operators are thus defined as learnable functions over pre-computed dilated spiral sequences [16]. Pooling and un-pooling operators are defined as sparse matrix multiplications with pre-computed transformations that are obtained with a quadric sampling procedure [16, 35] (see Supplementary Materials).

Our 3D-VAE is built as an encoder-decoder pair (Fig. 1, *Centre*), where the decoder is used as a generative model and is referred to as generator. Following this convention, we define our architecture as a pair of non-linear functions $\{E, G\}$. Let $\mathcal{X}$ be the vertex embedding domain and $\mathcal{Z}$ the latent distribution domain, we have $E : \mathcal{X} \rightarrow \mathcal{Z}$ defined as a variational distribution $q(\mathbf{z}|\mathbf{X})$ that approximates the intractable model posterior distribution, and $G : \mathcal{Z} \rightarrow \mathcal{X}$ described by the likelihood $p(\mathbf{X}|\mathbf{z})$. Throughout the entire network, each spiral++ convolutional layer is followed by an ELU activation function. However, in E convolutions are interleaved with pooling layers and in G by un-pooling layers. There are also three fully connected layers: two of them are the last layers of E predicting the mean and the diagonal covariance of the variational distribution, the other is the first layer of G and transforms $\mathbf{z} \sim \mathcal{Z}$ back into a low-dimensional mesh that can be processed by mesh convolutions.

During training, the following loss is minimised:

$$\mathcal{L}_{VAE} = \mathcal{L}_R + \alpha\mathcal{L}_L + \beta\mathcal{L}_{KL} \tag{1}$$

where α and β are weighting constants. $\mathcal{L}_R = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}'_n - \mathbf{x}_n\|_2^2$ is the mean squared error between the input ($\mathbf{x}_n \in \mathbf{X}$) and the corresponding output ($\mathbf{x}'_n \in \mathbf{X}' = G(E(\mathbf{X})) = G(\mathbf{z})$) vertices. This reconstruction loss encourages the output of the VAE to be as close as possible to its input. $\mathcal{L}_{KL} = KL[q(\mathbf{z}|\mathbf{X})||p(\mathbf{z})]$ is the Kullback–Leibler (KL) divergence pushing the variational distribution towards the prior distribution $p(\mathbf{z})$, which is defined as a standard spherical Gaussian distribution. Finally, $\mathcal{L}_L$ is a smoothing term based on the uniform Laplacian [30] that is computed on the output vertices as:

$$\mathcal{L}_L = \frac{1}{N}\sum_{n=1}^{N} \|\delta_n\|_2 \quad \text{with } \delta_n = \frac{1}{|\mathcal{N}_n|}\sum_{e\in\mathcal{N}_n} \mathbf{x}'_e - \mathbf{x}'_n$$

where δ_n is the Laplacian of the n-th output vertex, and $\mathcal{N}_n$ the set of its neighbouring vertices with cardinality $|\mathcal{N}_n|$. $\mathcal{L}_L$ is efficiently computed by relying on matrix operators. Concretely, we have $\boldsymbol{\Delta} = [\delta_1, ..., \delta_N]^T = \mathbf{L}\mathbf{X}'$, where $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ is the Laplacian operator with random walk normalisation, $\mathbf{A} \in \mathbb{N}^{N\times N}$ is the adjacency matrix and $\mathbf{D} \in \mathbb{R}^{N\times N}$ is the diagonal degree matrix with $D_{aa} = \sum_a A_{ab}$. Note that vertices are normalised by subtracting the per-vertex mean of the training set and dividing the result by the per-vertex standard deviation of the training set, thus the losses in Eq. 1 are computed on normalised vertices. Also, all loss terms are reduced across mini-batches with a mean reduction.

3.2. Mini-Batch Feature Swapping and Latent Consistency Loss

We aim to obtain a generative model where vertices corresponding to specific mesh features are controlled by a predefined set of latent variables. Therefore, we start by defining F arbitrary mesh features on a mesh template (Fig. 1, *Right*). Features are manually defined by colouring mesh vertices. Since vertices have point-wise correspondences (Sec. 3.1), features can be easily identified for every other mesh in the dataset without manually segmenting them. This allows us to swap features from one mesh to another by replacing the vertices corresponding to the selected feature (Fig. 2).

Feature swapping is at the core of our method and it allows us to curate the mini-batch generation in order to properly shape and constrain the latent representation of each

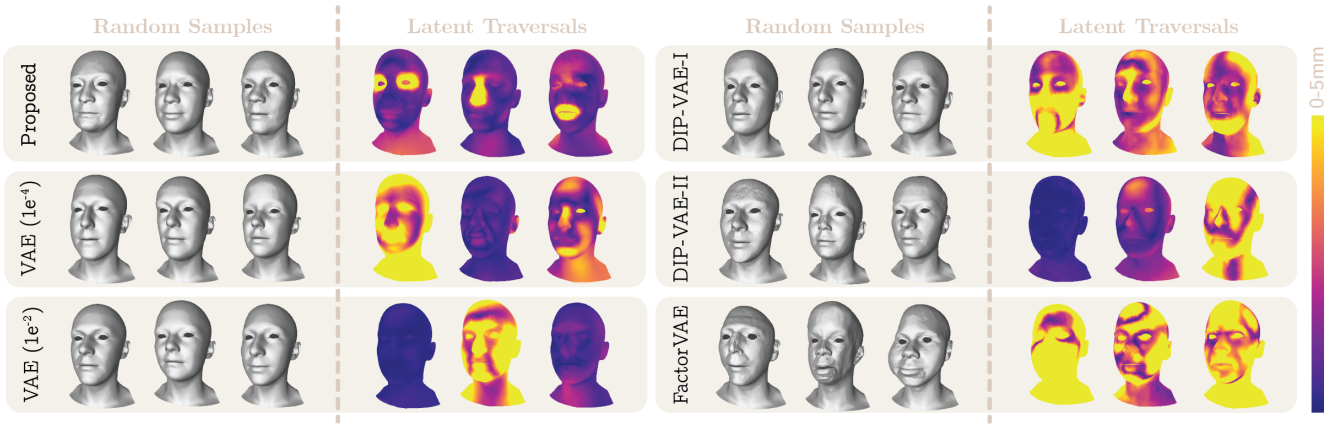

Figure 3. Random samples and vertex-wise distances showing the effects of traversing three randomly selected latent variables (see Supplementary Material to observe the effects for all the latent variables.)

mesh. Each mini-batch of size B can be thought of as a squared matrix of size $\sqrt{B} \times \sqrt{B}$, where each element $\mathbf{X}_{ij}$ is the vertex embedding of a different mesh. As it can be seen from Fig. 1 (*Left*), while elements on the diagonal of this matrix are loaded from the dataset, the remaining elements are created online by swapping features. Every time a mini-batch is created, a feature is randomly selected and swapped. Therefore, each row of the matrix contains the same mesh with different features, while each column contains different meshes with the same feature. Interestingly, the naive implementation of the feature swapping causes visible surface discontinuities in most input meshes (Fig. 2), but discontinuities are not present in reconstructed meshes thanks to the Laplacian regulariser in Eq. 1.

Obviously, when a mini-batch is encoded we obtain a batched latent. As we can see in Fig. 1 (*Centre*), for each $\mathbf{X}_{ij}$ we have a corresponding $\mathbf{z}_{ij} \sim E(\mathbf{X}_{ij})$ which is evenly split in F subsets of latent variables, one for each mesh feature ($\mathbf{z}_{ij} = \{\mathbf{z}^{\omega}_{ij}\}^{F}_{\omega=1}$). Note that even though every latent subset $\mathbf{z}^{\omega}_{ij}$ has the same number of variables, uneven splits are also admissible.

Every time a mini-batch is created by swapping a feature f, we can define $\mathbf{z}_{ij} = (\mathbf{z}^{f}_{ij}|\mathbf{z}^{c}_{ij})$. $\mathbf{z}^{f}_{ij}$ is the subset of latent variables controlling the feature swapped across the current mini-batch. $\mathbf{z}^{c}_{ij}$ is the part that controls everything else and is defined as $\mathbf{z}^{c}_{ij} = \{\mathbf{z}^{\omega}_{ij}\}^{F}_{\omega=1} \setminus \{\mathbf{z}^{f}_{ij}\}$. Inspired by both triplet losses and [37], and thanks to our curated mini-batching, we can enforce differences and similarities in the latent representation of the different $\mathbf{X}_{ij}$ by requiring matched $\mathbf{z}^{\omega}_{ij}$ pairs to have a distance in latent space that is smaller by a margin, η, than the distance for unmatched pairs. We traverse the diagonal of the mini-batch latent matrix and compare all the elements on the row containing the diagonal element $\mathbf{z}_{ss}$ with those in the column containing $\mathbf{z}_{ss}$ ($\forall s \in \{1, ..., \sqrt{B}\}$). When considering $\mathbf{z}^{f}_{ij}$ we enforce latent similarities across columns and latent differences across rows by evaluating: $\|\mathbf{z}^{f}_{ps} - \mathbf{z}^{f}_{qs}\|^2_2 + \eta_1 \leq \|\mathbf{z}^{f}_{sp} - \mathbf{z}^{f}_{sq}\|^2_2$, $\forall s, p, q \in \{1, ..., \sqrt{B}\}$ with $p \neq q$. This is justified by the fact that elements in $\mathbf{X}_{ij}$ have the same mesh feature across columns and different mesh features across rows. Vice versa, when considering $\mathbf{z}^{c}_{ij}$, which controls all the other mesh features for the current mini-batch, we enforce similarities row-wise and differences column-wise by evaluating: $\|\mathbf{z}^{c}_{sp} - \mathbf{z}^{c}_{sq}\|^2_2 + \eta_2 \leq \|\mathbf{z}^{c}_{ps} - \mathbf{z}^{c}_{qs}\|^2_2$, $\forall s, p, q \in \{1, ..., \sqrt{B}\}$ with $p \neq q$. We thus define our latent consistency loss as:

$$\mathcal{L}_c = \gamma \sum_{\substack{s,p,q=1 \\ p \neq q}}^{\sqrt{B}} \max\left[0, \|\mathbf{z}^{f}_{ps} - \mathbf{z}^{f}_{qs}\|^2_2 - \|\mathbf{z}^{f}_{sp} - \mathbf{z}^{f}_{sq}\|^2_2 + \eta_1\right] + \\ + \max\left[0, \|\mathbf{z}^{c}_{sp} - \mathbf{z}^{c}_{sq}\|^2_2 - \|\mathbf{z}^{c}_{ps} - \mathbf{z}^{c}_{qs}\|^2_2 + \eta_2\right] \tag{2}$$

where $\gamma = \frac{1}{B\sqrt{B}-B}$ is a batch normalisation term that considers all the latent distances comparisons performed while computing $\mathcal{L}_c$. Combining Eq. 1 with Eq. 2 and said $\kappa \in \mathbb{R}$ a weighting coefficient, we can formulate the total loss as:

$$\mathcal{L} = \mathcal{L}_{VAE} + \kappa\mathcal{L}_c = \mathcal{L}_R + \alpha\mathcal{L}_L + \beta\mathcal{L}_{KL} + \kappa\mathcal{L}_c \tag{3}$$

4. Experiments

Datasets Our main objective is to train a generative model capable of generating different identities from a set of feature-disentangled latent variables. For our experiments we require datasets containing as many subjects as possible in a neutral expression. However, most open source datasets for 3D shapes of faces, bodies, or animals contain only a limited number of subjects captured in different expressions or poses (e.g. MPI-Dyna [34], SMPL [28],

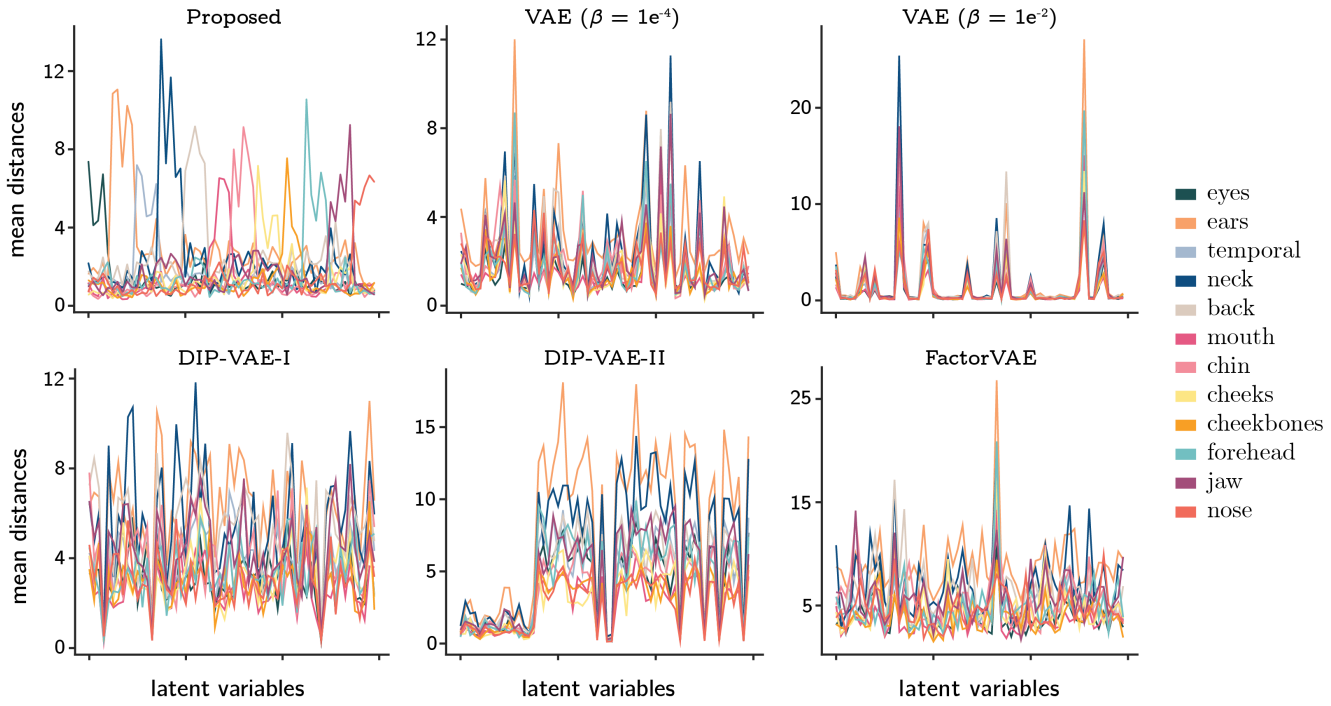

Figure 4. Effects of traversing each latent variable across different mesh features. For each latent variable (abscissas) we represent the per-feature mean distances computed after traversing the latent variable from its minimum to its maximum value. For each latent variable, we expect a high mean distance in one single feature and low values for all the others.

SURREAL [39], COMA [35], SMAL [47], etc.). For this reason, we rely on two linear models that were built using a conspicuous number of subjects and that are released for non-commercial scientific research purposes: UHM [33] and STAR [31] (Sec. 2). From these models we randomly generate $10,000$ meshes and create one dataset for faces and one for bodies. We use 90% of the data for training, 5% for validation, and 5% for testing.

Implementation Details All networks were implemented in PyTorch and trained for 40 epochs using the ADAM optimiser [20] with a fixed learning rate of $lr = 1e^{-4}$ and mini-batch size $B = 16$ (note that the feature swapping is applied to our method only). Spiral convolutions [1] had spiral length of 9 and spiral dilation of 1. The last convolutional layer of E and the first of G had 64 features while all the others 32. The sampling factors used during the quadric sampling for the creation of the up- and down-sampling transformation matrices were set to 4. Since the two datasets have a significantly different number of vertices ($N_{faces} = 71,928$ and $N_{bodies} = 6,890$), networks operating on faces have 4 convolutional layers interleaved with sampling operators in both E and G, while networks operating on bodies have only 3. For the same reason latent sizes are different: 60 variables for faces and 33 for bodies. Considering that the face template was segmented in 12 regions and the body template in 11, each $\mathbf{z}_{ij}^{\omega}$ has 5 variables for faces and 3 for bodies. The weight of the Laplacian regulariser was set to $\alpha = 1$, while the latent consistency weight was $\kappa = 0.5$ for faces and $\kappa = 1$ for bodies. η_1 and η_2 were set to $\eta_1 = \eta_2 = 0.5$ for both datasets. Training was performed on a single Nvidia Quadro P5000 for faces and on an Nvidia GeForce GTX 1050Ti for bodies. We run approximately 120 experiments in 25 GPU days.

Comparison with Other Methods We compare our method with other self-supervised methods based on encoder-decoder pairs. For a fair comparison, all methods share the same underlying architecture, which we refer to as VAE and which is already detailed in Sec. 3.1. Consistently with the current literature [14, 26, 35, 44], we found that the weight coefficient (β) on the KL divergence in VAEs for meshes is smaller than the one used for images. In fact, with $\beta \geq 1$ the VAE is not able to reconstruct the data. Thus, we report results on VAEs with $\beta \in \{1e^{-2}, 1e^{-4}\}$. It is worth noting that the discrepancy between meshes and images does not allow to define a β-VAE with the same criteria used in the literature ($\beta > 1$) [18]. We also compare

[1] The SpiralNet++ implementation was made available with an MIT license.

Table 1. Quantitative comparison between our model and other state-of-the-art methods for self-supervised latent disentanglement. All methods were trained on the same face dataset. Mean and Max Rec. refer to the the mean and maximum average per-vertex reconstruction errors across the test set. Values are computed in millimetres. The diversity is computed as detailed in Sec. 4. All the other metrics for the evaluation of the generation capabilities were introduced in [42].

Method	Mean Rec. (↓)	Max Rec. (↓)	Diversity (↑)	JSD (↓)	MMD (↓)		COV(%, ↑)		1-NNA (Δ%, ↓)	
					CD	EMD	CD	EMD	CD	EMD
VAE ($\beta = 1e^{-2}$)	1.47	1.99	5.43	**1.55**	1.66	0.43	62.99	63.67	7.25	7.50
VAE ($\beta = 1e^{-4}$)	**0.61**	**0.74**	4.23	4.89	1.53	0.38	65.49	**66.33**	1.17	**0.17**
DIP-VAE-I	4.65	11.86	4.74	5.32	**1.24**	**0.29**	55.57	56.42	4.31	4.56
DIP-VAE-II	4.76	11.92	4.30	6.44	1.70	0.43	48.48	47.30	17.72	17.15
Factor VAE	0.74	1.01	**10.51**	12.47	3.60	0.97	41.05	41.05	2.28	2.62
Proposed	0.73	0.93	4.23	4.30	1.56	0.38	**65.67**	63.67	**0.50**	1.67

our method with the DIP-VAE-I, DIP-VAE-II, and Factor VAE. To the best of our knowledge this is the first attempt to use them in the mesh domain. Therefore, for the two DIP-VAEs, we set $\beta = 1e^{-4}$ and, following the hyperparameter tuning strategy adopted in the original implementation [23], we tune λ_d and λ_{od}. Here we report results for DIP-VAE-I with $\lambda_d = 100$ and $\lambda_{od} = 10$ as well as for DIP-VAE-II with $\lambda_d = 10$ and $\lambda_{od} = 10$, which qualitatively showed better disentanglement performances. Factor VAE is trained with a discriminator learning rate of $1e^{-6}$, and a total correlation weight $\gamma = 0.25$.

We first evaluate the quality of the different models trained on the face dataset in terms of reconstruction errors, diversity of the generated samples, Jensen-Shannon Divergence (JSD) [2], Coverage (COV) [2], Minimum matching distance (MMD) [2], and 1-nearest neighbour accuracy (1-NNA) [42] (Tab. 1). Mean and maximum reconstruction errors are computed with respect to mean per-vertex errors across the test set. The diversity is computed as the mean of mean per-vertex distances among pairs of meshes randomly generated with the model. The other metrics are computed by leveraging the Chamfer (CD) and Earth Mover (EMD) distances on 2048 randomly selected pairs of vertices. Note that since the original formulation of 1-NNA expects scores converging to 50%, in Tab. 1 we report absolute differences between the original score and the 50% target value. From Tab. 1 we observe that while most methods for latent disentanglement have significantly increased reconstruction errors, our method closely match the VAE. We also notice that while most models have similar diversity, Factor VAE is able to generate more diverse data. While this property seems to be desirable, observing some randomly generated sample (Fig. 3), we argue that sampled faces are less realistic. The other metrics used to evaluate the generation capabilities of the different models show that our method is comparable with the others, thus proving that our mini-batching procedure and latent consistency loss do not negatively affect the generation capabilities.

Evaluation of Latent Disentanglement Previous work evaluated the latent disentanglement on either datasets where labelled data were available or on custom-made datasets of images whose generative factors could be used as labels. Examples of such datasets [18, 19, 23] are binary images of geometric shapes (e.g. circles, rectangles, etc.) where shape deformation parameters are known, or images rendered with controlled camera and lighting positions. Even though both our datasets are generated from existing models, these models lack control over the generative factors, thus traditional metrics such as Z-Diff [18], SAP [23], and Factor [19] scores cannot be computed. In addition, the few unsupervised disentanglement metrics currently existing [45], are not suitable for our evaluation because [27] is tailored for the evaluation of the disentanglement of style and content information, while [11] is used for model and hyperparameter selection thus requiring multiple computationally expensive hyperparameter sweeps. Therefore, we decide to evaluate the effects caused on the generated meshes while traversing each latent variable. We generate two meshes corresponding to each latent variable: one is created setting one latent variable to its minimum (-3) and all the remaining to their mean value (0), the other replacing the minimum with the maximum value ($+3$). The per-vertex Euclidean distances between the two shapes represent the effects of perturbing a single latent variable. These effects can be qualitatively assessed by observing meshes rendered with vertex colours proportional to the distances (Fig. 3, and Fig. 5 D). Alternatively, distances corresponding to each feature (Fig. 1, *Right* and Fig. 5 A) can be averaged and subsequently plotted as in Fig. 4 and Fig. 5 C. This representation clearly highlights how perturbing each latent variable affects the different features. While most methods appear to be difficult to interpret and mostly entangled, our method shows a significantly more structured, interpretable, and disentangled latent representation than other methods. Interestingly, in the VAE with $\beta = 1e^{-2}$ we observe a polarised regime in which only

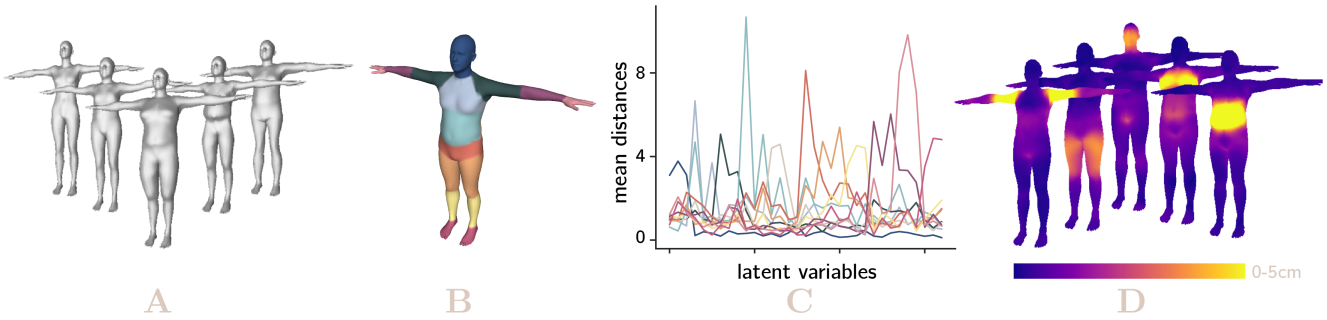

Figure 5. Results of our method on bodies. A: samples randomly generated with the proposed method trained on body meshes. B: visual representation of all the different body features for which we seek to obtain a disentangled latent representation. C: effects of latent variable traversals for each latent variable across different body features. D: vertex-wise distances showing the effects of traversing five latent variables (see Supplementary Material for all the latent variables).

a subset of latent variables control the generated shapes. However, these variables are also controlling the same features, thus disentanglement is not achieved. Since the polarised regime occurs in β-VAEs [36], we can consider this VAE to be a β-VAE operating on meshes.

Direct Manipulation Similarly to [17], our method supports direct manipulation of the generated 3D meshes. A user is thus able to select one or multiple vertices, specify their new desired location, and our method automatically generates a new mesh locally deformed to satisfy the user edit. This is achieved through a small optimisation procedure over the latent representation. We use the ADAM optimiser for 50 iterations and with a fixed learning rate of $lr = 0.1$. Given $S \circ \mathbf{X}' = S \circ G(\mathbf{z}) \in \mathbb{R}^{\Upsilon \times 3}$ the subset of vertices manually selected from the currently generated mesh, and their desired positions $\mathbf{Y} \in \mathbb{R}^{\Upsilon \times 3}$, with Υ representing the number of selected vertices, we optimise: $\min_{\mathbf{z}^f} \|S \circ G(\mathbf{z}) - \mathbf{Y}\|_2^2$. Note that the optimisation over $\mathbf{z}^f$ guarantees the locality of the manipulation (Fig. 6, IIa) and it is achieved by setting to zero the gradients computed over $\mathbf{z}^c$. This is made possible by our method and its improved latent disentanglement. An optimisation over the entire latent representation would cause visible global changes (Fig. 6, IIb), thus making impossible the direct manipulation.

5. Conclusion

We proposed a novel approach to learn a more disentangled, interpretable, and structured latent representation for 3D VAEs. This is achieved by curating the mini-batching procedure with feature swapping and introducing an additional latent consistency loss. Even though our method is able to disentangle predefined subsets of latent variables, we do not guarantee orthogonality and disentanglement among the variables within each subsets. Nonetheless, we can increase the number of subsets to achieve finer control over the generated model. The main limitations of our work are the assumptions made on the training data. A consistent scaling and alignment, as well as dense-point correspondences, and a fixed mesh topology are common for generative models of 3D faces (and bodies) and useful for an efficient feature swapping. However, this assumption could be relaxed to make our method suitable for more general 3D problems if a different architecture was implemented and semantic segmentations of each 3D shape were available. Good semantic segmentations are not trivial to obtain for raw data, but methods such as [40, 41] could be used. As future work, we aim at introducing and properly disentangling expressions (or poses) while retaining the superior latent disentanglement over identity features made possible by our method.

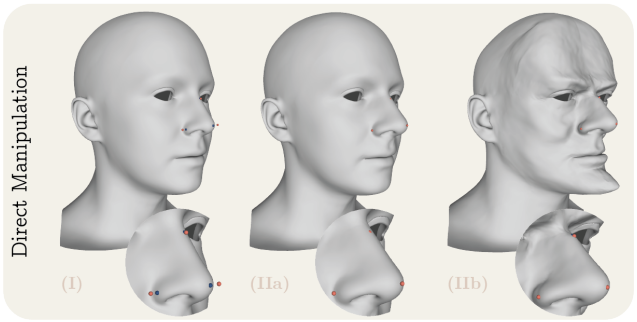

Figure 6. Direct manipulation of the generated mesh. (I) The user selects an arbitrary number of vertices (blue) and their new desired position (red), then our method generates a locally edited mesh fitting the desired locations. Results are reported optimising only $\mathbf{z}^f$ (IIa) and optimising the entire $\mathbf{z}$ (IIb).

Acknowledgement This work was supported by the Wellcome Trust/EPSRC [203145Z/16/Z]. The views expressed in this publication are those of the author(s) and not necessarily those of the Wellcome Trust.

References

[1] Victoria Fernández Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, and Edmond Boyer. A decoupled 3d facial shape model by adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9419–9428, 2019. 2, 3

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 7

[3] Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. Geometric disentanglement for generative latent shape models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8181–8190, 2019. 2, 3

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[5] 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. 2

[6] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, 2016. 2

[7] Shiyang Cheng, Michael Bronstein, Yuxiang Zhou, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. Meshgan: Non-linear 3d morphable models of faces. *arXiv preprint arXiv:1903.10384*, 2019. 2, 3

[8] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodola. Limp: Learning latent shape representations with metric preservation priors. *arXiv preprint arXiv:2003.12283*, 2, 2020. 2, 3

[9] Hang Dai, Nick Pears, William Smith, and Christian Duncan. Statistical modeling of craniofacial shape and texture. *International Journal of Computer Vision*, 128(2):547–571, 2020. 2

[10] Zheng Ding, Yifan Xu, Weijian Xu, Gaurav Parmar, Yang Yang, Max Welling, and Zhuowen Tu. Guided variational autoencoder for disentanglement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7920–7929, 2020. 2, 3

[11] Sunny Duan, Loic Matthey, Andre Saraiva, Nicholas Watters, Christopher P Burgess, Alexander Lerchner, and Irina Higgins. Unsupervised model selection for variational disentangled representation learning. *arXiv preprint arXiv:1905.12614*, 2019. 7

[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. 2

[13] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2525–2534. PMLR, 2019. 2, 3

[14] Simone Foti, Bongjin Koo, Thomas Dowrick, Joao Ramalhinho, Moustafa Allam, Brian Davidson, Danail Stoyanov, and Matthew J Clarkson. Intraoperative liver surface completion with graph convolutional vae. In *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis*, pages 198–207. Springer, 2020. 6

[15] Baris Gecer, Alexandros Lattas, Stylianos Ploumpis, Jiankang Deng, Athanasios Papaioannou, Stylianos Moschoglou, and Stefanos Zafeiriou. Synthesizing coupled 3d face modalities by trunk-branch generative adversarial networks. In *European Conference on Computer Vision*, pages 415–433. Springer, 2020. 2, 3

[16] 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. 2, 3, 4

[17] Aurel Gruber, Marco Fratarcangeli, Gaspard Zoss, Roman Cattaneo, Thabo Beeler, Markus Gross, and Derek Bradley. Interactive sculpting of digital faces using an anatomical modeling paradigm. In *Computer Graphics Forum*, volume 39, pages 93–102. Wiley Online Library, 2020. 2, 3, 8

[18] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2016. 2, 3, 6, 7

[19] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 2, 3, 7

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[22] Tejas D Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B Tenenbaum. Deep convolutional inverse graphics network. *arXiv preprint arXiv:1503.03167*, 2015. 2, 3

[23] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017. 3, 7

[24] Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xiang, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, et al. Learning formation of physically-based face attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3410–3419, 2020. 2, 3

[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 Trans. Graph.*, 36(6):194–1, 2017. 2

[26] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1886–1895, 2018. 3, 6

[27] Xiao Liu, Spyridon Thermos, Gabriele Valvano, Agisilaos Chartsias, Alison O'Neil, and Sotirios A Tsaftaris. Metrics for exposing the biases of content-style disentanglement. *arXiv preprint arXiv:2008.12378*, 2020. 7

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2, 3, 5

[29] Charlie Nash and Christopher KI Williams. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. In *Computer Graphics Forum*, volume 36, pages 1–12. Wiley Online Library, 2017. 2, 3

[30] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389, 2006. 4

[31] Ahmed AA Osman, Timo Bolkart, and Michael J Black. Star: Sparse trained articulated human body regressor. *arXiv preprint arXiv:2008.08535*, 2020. 2, 3, 6

[32] Stylianos Ploumpis, Evangelos Ververas, Eimear O'Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick Pears, William Smith, Baris Gecer, and Stefanos P Zafeiriou. Towards a complete 3d morphable model of the human head. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[33] Stylianos Ploumpis, Haoyang Wang, Nick Pears, William AP Smith, and Stefanos Zafeiriou. Combining 3d morphable models: A large scale face-and-head model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10934–10943, 2019. 2, 3, 6

[34] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)*, 34(4):1–14, 2015. 5

[35] 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. 2, 3, 4, 6

[36] Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12406–12415, 2019. 8

[37] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7763–7772, 2019. 5

[38] J Rafael Tena, Fernando De la Torre, and Iain Matthews. Interactive region-based linear 3d face models. In *ACM SIGGRAPH 2011 papers*, pages 1–10. Association for Computing Machinery, 2011. 2

[39] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. 6

[40] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2598–2606, 2018. 8

[41] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 8

[42] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 7

[43] Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J Guibas, and Lin Gao. Dsm-net: Disentangled structured mesh net for controllable generation of fine geometry. *arXiv preprint arXiv:2008.05440*, 2020. 2, 3

[44] Yu-Jie Yuan, Yu-Kun Lai, Jie Yang, Qi Duan, Hongbo Fu, and Lin Gao. Mesh variational autoencoders with edge contraction pooling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 274–275, 2020. 3, 6

[45] Julian Zaidi, Jonathan Boilard, Ghyslain Gagnon, and Marc-André Carbonneau. Measuring disentanglement: A review of metrics. *arXiv preprint arXiv:2012.09276*, 2020. 7

[46] Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. Fully convolutional mesh autoencoder using efficient spatially varying kernels. *arXiv preprint arXiv:2006.04325*, 2020. 3

[47] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017. 6