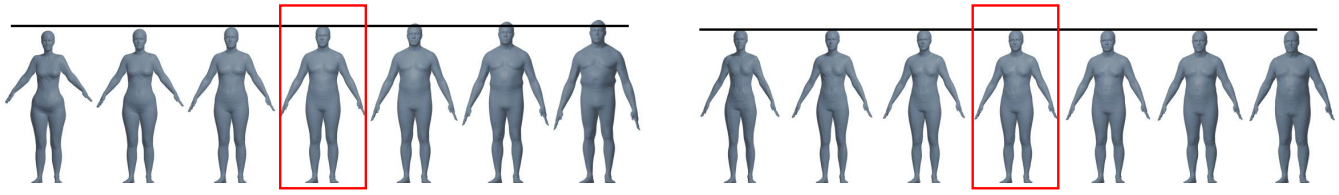
RightSizing: Disentangling Generative Models of Human Body Shapes with Metric Constraints

Yuhao Wu
wuyuhao@cs.ubc.ca
Department of Computer Science,
University of British Columbia
Vancouver, BC, Canada

Chang Shu
Chang.Shu@nrc-cnrc.gc.ca
National Research Council Canada
Ottawa, ON, Canada

Dinesh K. Pai
Pai@cs.ubc.ca
Department of Computer Science,
University of British Columbia
Vancouver, BC, Canada



(a) Changing waist girth with a standard method, PCA with feature analysis

(b) Changing waist girth with our method, RightSizing

Figure 1: With current generative models, it is difficult to modify one feature (e.g., waist girth) without also affecting other features such as height. Part (a) shows the result of a widely used classical method, PCA analysis with feature analysis (PCA-FA) [Allen et al. 2003]; recent methods have similar flaws as we demonstrate here. With our method (b) we can modify the waist without affecting height. See horizontal bars above the figures. More generally, our method can disentangle multiple features so that they can be controlled independently, leaving other features unchanged. This greatly simplifies the use of generative models in computer graphics.

ABSTRACT

Deep generative models such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and diffusion models have demonstrated their efficacy in generating 2D images and 3D meshes. However, interpreting and controlling the learned latent space is very difficult, severely limiting the utility of these methods. Worse, it has been shown that fully disentangling the latent space using only unsupervised methods is theoretically infeasible.

In this work, we introduce a novel method for latent space disentanglement on 3D meshes that achieves interpretability, control, and strong disentanglement. Our method comprises two components: a learned feature function for predicting 3D mesh features, and a generative model that predicts not only the desired meshes but also their features and feature gradients. We employ feature gradients as part of the loss function to promote disentanglement. Experimental results demonstrate that our disentanglement method is highly effective and achieves strong disentanglement without compromising the accuracy of the reconstruction.

Graphics Interface 2024,
,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS CONCEPTS

• **Computing methodologies** → **Shape analysis**; **Mesh geometry models**; **Modeling and simulation**.

KEYWORDS

human shape generation, disentanglement, generative models, latent representation, human body sizing

ACM Reference Format:
Yuhao Wu, Chang Shu, and Dinesh K. Pai. 2024. RightSizing: Disentangling Generative Models of Human Body Shapes with Metric Constraints. In *Proceedings of Graphics Interface 2024.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Learning latent representations of 3D meshes is an effective technique for understanding the shape space of the human body. It compresses data by representing complex 3D models as compact vectors, drastically reducing storage needs while maintaining high fidelity during reconstruction. With these representations, new shapes can be generated by sampling the latent space. In applications, it enables rapid prototyping for product design, product customization to fit human shape, and personalized avatars in virtual worlds. A simple and widely used way to obtain latent representations is principal component analysis (PCA). It achieves a reduction in dimensionality by extracting orthogonal components, representing the most significant sources of variability in high-dimensional data while ensuring their independence. In recent years, there has been significant interest in using deep generative models to learn latent representations [Goodfellow et al. 2014; Kumar et al. 2018;

Sohl-Dickstein et al. 2015]. These models tend to have fewer parameters and have been demonstrated to achieve better reconstruction accuracy and may generalize to larger deformations.

For most graphics applications, however, it is not random shape generation that is useful but we often want to generate shapes with certain properties. This requires the latent variables to be interpretable and meaningful. Recent efforts have aimed to "disentangle" the latent space to improve interpretability. Although there is no universally accepted formal definition of disentanglement, the consensus is that such a representation should isolate different data variation factors. Specifically, a change in one factor of variation should result in a change in just one component of the learned representation [Bengio et al. 2013; Locatello 2021].

One group of disentanglement work focused on **discovering** meaningful factors and adopted unsupervised approaches [Burgess et al. 2018; Chen et al. 2018; Eastwood and Williams 2018; Higgins et al. 2016; Kim and Mnih 2018; Kumar et al. 2018; Ridgeway and Mozer 2018]. For VAEs, these approaches share the fundamental idea of decomposing the KL divergence loss to isolate a total correlation term, which encourages independent latent variables. These models are valuable when we do not know much about the dataset and want to explore its variability.

Another group of work considered **designing** the structure of the latent space. We often have some knowledge of our data and want to control the shape generation process. In general, it is difficult to label data to supervise the disentanglement, but we can often weakly supervise training by arranging data in groups that isolate certain factors [Zhou et al. 2020a] or go completely unsupervised by exploiting the intrinsic properties of data [Aumentado-Armstrong et al. 2019; Cosmo et al. 2020; Yang et al. 2023b].

Both groups have limitations. The discovery approach cannot control the semantic meaning of each latent dimension, a serious flaw in many computer graphics applications that require control of body shape. Therefore, the latent design approach is favored. However, current methods are limited to few types of disentanglement, such as separating pose and shape. Disentanglement of shape variation based on general measurable characteristics of shape is largely unexplored.

In this paper, we propose a novel approach, focusing on the important case of anthropometric features of 3D human body meshes, though our approach is more general. Our main contributions are:

- A systematic procedure for designing disentangled latent spaces with separate user-specified metric features. It provides clear semantics and control of generative models.
- The latent space gradient method, which enforces strong disentanglement of the latent space using feature gradients.
- Learned feature functions that predict user-specified metric features of the body from a given mesh with high accuracy, as well as their first and second derivatives. We show how automatic differentiation tools can be leveraged to compute higher order derivatives efficiently.
- Experimental results based on real human scan data that demonstrate strong disentanglement, control, and interpretability, while preserving good reconstruction quality.

The source code associated with this paper is available on Github at *https://github.com/ai4d/RightSizing*.

2 RELATED WORK

The literature on 3D shape analysis, generation, and disentanglement is extensive. Here, we review the mesh-based representations with an emphasis on human shape modeling.

2.1 3D Shape Analysis and Generation

2.1.1 Linear Statistical Models. Blanz and Vetter [1999] pioneered the use of PCA to build a statistical model to represent 3D face scans. This work laid the foundation for a series of PCA-related methods for modeling 3D human faces and bodies [Allen et al. 2003; Amberg et al. 2008; Thies et al. 2016; Yang et al. 2011]. Allen et al. [2003] deforms a template mesh to human body scans to register the shape data and apply PCA to the vertices of the meshes. The SCAPE model [Anguelov et al. 2005] includes both body shape and pose by modeling triangle-based deformation. This was further improved by the SMPL model [Loper et al. 2015] which parameterizes an explicit skeletal structure with linear blend skinning. The SMPL model greatly improved shape generation efficiency by modeling vertex-based deformation. It is now widely used in many applications [Bogo et al. 2016; Kanazawa et al. 2018] and has also been extended to include hands and facial expressions [Pavlakos et al. 2019; Romero et al. 2017].

PCA-based linear models can be used to regress with body measurements, a technique called *feature analysis* [Allen et al. 2003]. Streuber et al. [2016] used this technique to relate body shape with words and generate human avatars using natural language descriptions. However, these methods cannot generate disentangled shapes, as we show in Figures 4 and Section 5.3.

Note that the open-source software MakeHuman [MakeHuman 2024] generates virtual characters from feature parameters such as age, gender, etc. However, there is little published information on how the system generates models, nor on the data used [Briceno and Paul 2019]. The system appears to generate models by interpolation and manually restricting the influence of the parameters.

2.1.2 Deep Generative Models. Graph convolutional networks extend the convolution from image to graph and are perfectly suitable for mesh. The CoMA framework [Ranjan et al. 2018] employed a ChebNet-based variational autoencoder to model face meshes with different expressions. As an alternative to spectral convolution layers, the Neural3DMM [Bouritsas et al. 2019] uses operators that do convolution in a spiral path around each vertex. Zhou et al. [2020b], introduced a distinctive convolution method in which each vertex was equipped with its own convolution kernel, thus transcending the constraints of a template-specific surface mesh. The MeshCNN architecture [Hanocka et al. 2019] learns the attributes of the edges and performs the pooling through an edge collapse mechanism.

2.2 Latent Space Disentanglement

2.2.1 Latent Space Discovery. Early work on disentangling the latent space of generative models focused on unsupervised methods [Burgess et al. 2018; Chen et al. 2018, 2016; Eastwood and Williams 2018; Higgins et al. 2016; Kim and Mnih 2018; Kumar et al. 2018; Ridgeway and Mozer 2018]. InfoGAN [Chen et al. 2016] discovers meaningful hidden representations in various image datasets by maximizing the mutual information between a subset of the

GAN's noise variables and the observations. β-VAE [Higgins et al. 2016] modifies the traditional VAE framework by introducing a tunable hyperparameter β, which effectively regulates the balance between latent channel capacity, independence constraints, and reconstruction accuracy.

Chen et al. [2018] improved the β-VAE model by identifying a term that quantifies the total correlation between latent variables through the decomposition of the evidence lower bound, which encourages the model to find statistically independent factors in the data distribution. The new model is called β-TCVAE.

Although unsupervised methods can sometimes provide surprising interpretations of the data, they do not always produce meaningful latent variables. Theoretical and experimental work shows that they often fail to find independent factors [Locatello et al. 2019].

2.2.2 Latent Space Design. A line of work in computer graphics and computer vision focused on factorizing the latent space into parts that correspond to different aspects of the data [Aumentado-Armstrong et al. 2019; Cosmo et al. 2020; Yang et al. 2023b; Zhou et al. 2020a]. These methods are motivated by applications, and they can be fully supervised [Yang et al. 2023a], weakly supervised by regrouping or pairing data [Cosmo et al. 2020; Jiang et al. 2019; Kulkarni et al. 2015; Zhou et al. 2020a], or unsupervised [Aumentado-Armstrong et al. 2019; Foti et al. 2023]. Most of the work on human modeling focused on disentangling pose and shape. These methods constrain the cost function using geometric properties. For example, Zhou et al. [2020a] uses *as rigid as possible* (ARAP) energy to impose self-consistency between the same subject in different poses. Another example is the GeoLatent method [Yang et al. 2023b], which uses a Riemannian metric to ensure that straight-line interpolations in latent codes follow geodesic curves and disentangle pose and shape variations at different scales. Recently, Sun et al. [2023] used skeleton information to separate pose from shape in an unsupervised way. They also proposed a method for editing individual body parts. However, since they used cylinders to represent the limbs and the torso, the shape controls may be limited. Aliari et al. [2023] trained a VAE model using segmented parts of the human face to achieve localized shape controls. Except for these last two methods, existing work tends to address the disentanglement of high-level information like pose and shape. In contrast, our method provides more flexible control by using a feature function, which can be obtained either through learning or computing it directly from the mesh.

3 METHODS

3.1 Overview

Let $\boldsymbol{x} \in \boldsymbol{X}$ represent the shape of a human body in the space of human shapes $\boldsymbol{X}$. If the training data are registered, as is common practice, to a template mesh with n vertices, $\boldsymbol{x}$ is an array of $3n$ vertex coordinates of the mesh. Let $\boldsymbol{z} \in \boldsymbol{Z}$ be an l-dimensional latent vector, with the i^{th} component designated z_i. Our goal is to construct a semantically meaningful latent space $\boldsymbol{Z}$ and a generative model $\hat{\boldsymbol{x}} = p(\boldsymbol{z})$ that can generate plausible samples $\hat{\boldsymbol{x}}$ from $\boldsymbol{z}$.

An important aspect of our approach is that meanings are not arbitrary or emergent, but can be designed to be m specific, human-interpretable features, denoted $\boldsymbol{h} \in \mathbb{R}^m$. We illustrate this approach with standard anthropometric features, such as height, weight, waist girth, etc.

In general, we will have more latent variables than features, i.e., $l \geq m$, since the latent space must be sufficiently rich to capture the variability in the data, and not just the features of interest. Furthermore, we may only want to control a subset of $c \leq m$ meaningful features. We can partition $\boldsymbol{Z}$ into a subspace $\boldsymbol{Z_c}$ that governs the controlled features of interest, and a subspace $\boldsymbol{Z_u}$ that governs the remaining features. We will refer to the former as the control variables and the latter as the uncontrolled variables. For simplicity of exposition, and without loss of generality, we will assume that latent variables and features are ordered with the first c components corresponding to the controlled variables, with z_i controlling h_i, for $i \in [0 \ldots c-1]$.

Our goal is to design a latent space with the following desiderata: (a) **interpretability**: specific control latent variables $z_i \in \boldsymbol{Z_c}$ are associated with specific quantitative features of the human body, h_i, such as height or waist girth; (b) **controllability**: we can vary the value of z_i to continuously modify the value of h_i; (c) **strong disentanglement**: the latent variable z_i does not affect other features $h_{j \neq i}$; and (d) **diversity**: the model $p(\boldsymbol{z})$ is capable of generating samples that are plausible and diverse with fixed values of some features h_i.

3.2 Feature Function

The features $\boldsymbol{h}$ depend on the body shape $\boldsymbol{x}$. Surprisingly, some anthropometric features, such as waist girth, are not directly computable from mesh data. For example, what is considered the "waist" depends on body type, pose, gender, and even intended use, and may need to be learned from measurements taken by human experts. Therefore we compute these features using a learned feature function $f : \boldsymbol{X} \rightarrow \mathbb{R}^m$. The feature function is a neural network that computes $\boldsymbol{h} = f(\boldsymbol{x})$ from the mesh data $\boldsymbol{x}$. The feature function is trained on a dataset of meshes with known features.

3.3 Feature Gradients in the Latent Space

We want h_j, the j^{th} feature, to be controlled only by the latent variable z_j and disentangled from all other variables $z_{i \neq j}$. Since both the model $p(\boldsymbol{z})$ and the feature function $f(\boldsymbol{x})$ are differentiable, we can compute the gradient of h_j with respect to latent vector $\boldsymbol{z}$ as $\nabla_{\boldsymbol{z}} h_j$. This gradient completely captures how the feature h_j changes as we vary the latent variables z_i locally around the current value of $\boldsymbol{z}$. It is the key to our approach to controllability and disentanglement. We achieve controllability if $\frac{\partial h_j}{\partial z_j} \neq 0$, and strong disentanglement by constraining the derivatives to be zero for all other variables, i.e., $\frac{\partial h_j}{\partial z_i} = 0$ for $i \neq j$. In Section 3.5, we describe how this is incorporated in the loss function.

3.4 Network Architecture

Figure 2 depicts the architecture of the proposed network. It comprises three fundamental components.

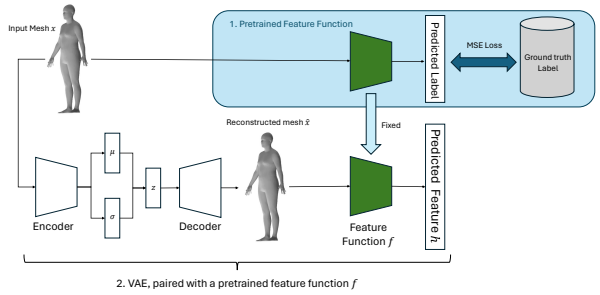


Figure 2: Architecture: (1) The feature network (top) uses a typical encoder structure to extract features h, such as height and weight, from raw mesh data x. (2) A traditional VAE, paired with the feature function (bottom) predicts h from latent vectors z. (3) Use the PyTorch auto-grad tools (not shown) to compute the gradient of the controlled features with respect to the latent variables, and use the gradient to disentangle the latent space.

The first component, the feature function, is a deep neural network that predicts the features $\boldsymbol{h} = f(\boldsymbol{x})$ from body shape $\boldsymbol{x}$. The features used in our work include height, chest circumference, waist circumference, hip circumference, arm length, and crotch height. The feature function is trained separately on a dataset of meshes with known features, prior to its use in the proposed method.

The second component is a traditional Variational Autoencoder (VAE) paired with the trained feature function network. The VAE consists of an encoder $q(\boldsymbol{z}|\boldsymbol{x})$ and a decoder $p(\boldsymbol{z})$ that work together. The encoder q takes an input mesh $\boldsymbol{x}$, and outputs the parameters of the latent distribution, μ and σ. Using the re-parameterization trick [Kingma and Welling 2014], we generate the distribution of the latent variable $\boldsymbol{z}$. The decoder takes a sample $\boldsymbol{z}$ as input and outputs the reconstructed mesh $\hat{\boldsymbol{x}}$. We follow the VAE with the feature function network $f(\boldsymbol{x})$.

The third component is autodiff-based gradient computation that computes gradients of all controlled features $\nabla_{\boldsymbol{z}} h_j$. The above network implicitly defines the composite function $\boldsymbol{h}(\boldsymbol{z}) = f(p(\boldsymbol{z}))$. We use this to compute the gradient of h_j with respect to $\boldsymbol{z}$. Note, however, that we need the gradient to be explicitly represented as a network (and not just computed as a side effect of training the VAE), since our disentanglement losses include the gradient. This gradient network (not shown in the Fig. 2) is efficiently constructed using built-in automatic differentiation tools in PyTorch.

3.5 Loss Functions

The loss function in our proposed method comprises several components, each representing a different requirement or "task". We minimize these losses using a multitask learning approach (Sec. 3.6).

- **VAE Losses**: The first two components are losses associated with the traditional VAE, and include the reconstruction loss and the KL divergence loss. Reconstruction loss, $L_r = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$ aims to preserve the information content of the input data. The KL divergence loss ensures that the latent space exhibits a structured and meaningful representation, denoted as $L_k = \mathrm{KL}(N(\mu, \sigma), N(0, 1))$.
- **Disentanglement Loss**: This is a novel aspect of our method. Our goal is to encourage strong disentanglement by penalizing derivatives of controlled features h_j to be zero with respect to other latent variables, i.e., $\frac{\partial h_j}{\partial z_i} = 0$ for $i \neq j$. More formally, we can define a projection matrix P^j onto the orthogonal complement of the basis vector $\boldsymbol{e}_j \in \boldsymbol{Z}$ by defining its k, l element as follows:

$$P^j_{k,l} = \begin{cases} 1, & \text{if } k = l \neq j \\ 0, & \text{otherwise.} \end{cases}$$

  Then, the loss for controlled feature h_j is $L^j_d = \|P^j \frac{\partial h_j}{\partial \boldsymbol{z}}\|^2$. In the face image generation literature, *contrastive learning* has been used to obtain orthogonality of the latent dimensions [Deng et al. 2020]. This method relies on sampling to penalize the entangled features. In contrast, our gradient-based method is simpler and more efficient.
- **Smoothness Loss**: The generated results occasionally exhibit high-frequency noise, leading to a bumpy appearance. To address this, we propose the addition of a simple regularization term to enhance the smoothness of the output. Specifically, we will use a loss based on the Laplace-Beltrami operator applied to the mesh which provides a simple measure of curvature. We use the Cotangent Laplacian [Meyer et al. 2003; Pinkall and Polthier 1993; Sorkine 2005] for this purpose. The Laplacian can not be directly used as a regularization term since body shapes have some natural high frequency features, particularly in the face and hands. Instead, we use the difference between the Laplacian of the generated mesh and a template mesh as the regularization term. The smoothness loss term is thus formulated as

$$L_s = \|\Delta(\hat{\boldsymbol{x}}) - \Delta(\boldsymbol{x}_T)\|^2,$$

  where $\Delta(\hat{\boldsymbol{x}})$ and $\Delta(\boldsymbol{x}_T)$ represent the Laplacians of the generated and template meshes, respectively.

3.6 Automatic weight-balancing

Multitask learning is a paradigm in machine learning where multiple learning tasks are solved simultaneously. Here, we borrow a weight-balancing technique from multi-task learning in order to assign appropriate relative weights to optimize a weighted combination of individual loss terms. The total loss may be defined as $L = \sum_{\tau \in \mathcal{T}} c_\tau \cdot L_\tau$, where L_τ are the losses defined in Sec. 3.5, and c_τ are the respective weights. For optimal results, it is essential to fine-tune the weights associated with each term.

To avoid manual weight adjustment, we implemented an algorithm that optimizes the weights during training [Kendall et al. 2018; Liebel and Körner 2018] . Specifically, the total loss is reformulated as:

$$L = \sum_{\tau \in \mathcal{T}} \frac{1}{2 \cdot c_\tau^2} \cdot L_\tau + \ln(1 + c_\tau^2). \tag{1}$$

Here, c_τ is now a learnable parameter, dynamically adjusted during the training process. This objective is then optimized using stochastic gradient descent.

4 IMPLEMENTATION DETAILS

4.1 Dataset

We used data from the CAESAR survey [Robinette et al. 2002]. Data were processed using the approach of [Xi et al. 2007] that fitted a templated model to each scan using anthropometric landmarks to guide the deformation, similar to [Allen et al. 2003]. For datasets that do not have landmarks, more advanced registration methods, such as CoRegistration [Hirshberg et al. 2012] can be used. We used 2,169 processed meshes, each with 20,000 faces and 10,002 vertices. In addition to meshes, the dataset provides anthropometric tables that include attributes like height, weight, and arm lengths, among others, measured by expert human measurers. We used these tables to learn the feature function f for the dataset.

In the datasets shown in the majority of disentanglement studies, such as *dSprites* [Higgins et al. 2016; Matthey et al. 2017], *Cars3D* [Reed et al. 2015], *SmallNORB* [LeCun et al. 2004], *Shapes3D* [Kim and Mnih 2018], each data sample x is obtained as a deterministic function of latent variable z. These datasets, often created artificially, resemble toy datasets designed for disentanglement tasks. By contrast, our dataset originates from real-world human and is not disentangled, e.g., we don't have data with just height varying.

4.2 Autoencoder Structure

Our primary objective lies in identifying an effective method for disentanglement and our method does not depend on a specific autoencoder network. Here we select CoMA as our base VAE. Our model's architecture, aligning with the CoMA model [Ranjan et al. 2018], sets the latent variable z's dimension at 8. As shown in Figure 3, our model includes four Chebyshev convolutional filters, each using $K = 6$ Chebyshev polynomials [Defferrard et al. 2016]. The encoder's convolutional layers feature output channels in the sequence of 16, 16, 16, and 32, while the decoder's layers have input channels in the sequence of 32, 16, 16, and 16.

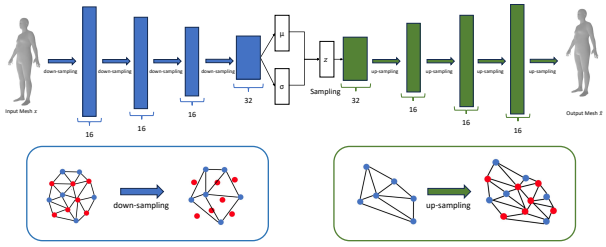


Figure 3: The network structure of Convolutional Mesh Autoencoder(CoMA).

Each Chebyshev layer in our model is paired with a down-sampling layer that reduces the vertex count by a factor of four. Similar to the uniformity of pixel dimensions in 2D CNN inputs, all meshes in our dataset share the same topology. We use the dataset's mean mesh as our template mesh. This template mesh undergoes downsampling via iterative contraction of vertex pairs, selectively removing vertices to minimize quadric error [Garland and Heckbert 1997]. During this process, the up-sampling matrix is also established. For all meshes in the dataset, our approach uses the same down-sampling and up-sampling matrices as the template.

4.3 Latent space gradients

Our method takes advantage of the automatic differentiation functionality of the Pytorch, which is as efficient as coding differentiation analytically, and less error prone. The loss function includes a term dependent on the first derivative of the feature function with respect to the latent variable z, requiring the use of its second derivative for optimization. The derivative of the feature function with respect to z is computed using the `torch.autograd.grad` interface. However, under standard conditions PyTorch frees intermediate buffers from the forward pass during the backward pass to optimize memory usage, effectively destroying the computation graph after the calculation of gradients. We ensure that the computational graph is retained, and can be reused for optimizing the disentanglement loss.

5 RESULTS

5.1 Feature Function Accuracy

We use a Graph Convolutional Network (GCN) as our predictor network to capture topological information of the body. This predictor network mirrors the encoder structure of CoMA [Ranjan et al. 2018]. We use the Mean Squared Error (MSE) loss between the predicted and ground truth labels. The network was trained over 50 epochs using the training dataset, and the results of the test set are presented in Table 1.

As shown in Table 1, our predictor network demonstrates a high level of accuracy in extracting features from a 3D human body mesh.

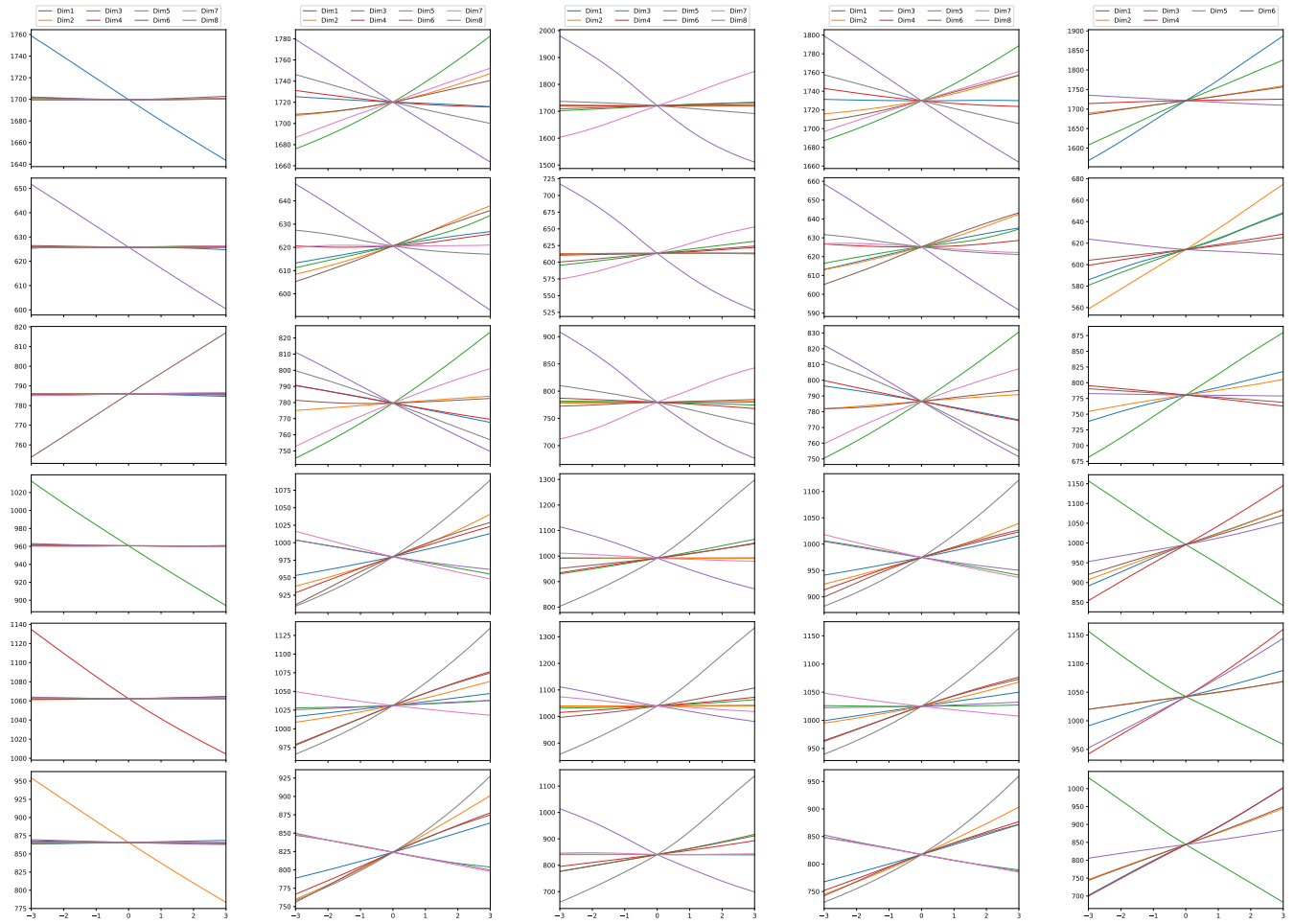
5.2 Reconstruction Quality

In our study, we calculated the reconstruction loss for VAE, β-VAE and β-TCVAE, in order to compare these methods with our proposed approach. The results are presented in Table 2.

Our method demonstrates superior reconstruction performance compared to other state-of-the-art VAE-based methods. Notably, relying solely on the reconstruction error might not provide a comprehensive understanding of the quality of reconstruction. As observed in Table 2, β-VAE exhibits higher reconstruction errors, compared to VAE and β-TCVAE, which show lower errors. However, visual inspections reveal that a lower reconstruction error can sometimes be associated with non-smooth features, indicative of high-frequency noise. To account for this limitation, we introduced an additional term: the smoothness loss.

The calculation of the smoothness loss follows the methodology outlined in Section 3.5. For each vertex $\mathbf{v}_i$, the Laplacian is defined as $(\Delta f)_i = \frac{1}{a_i} \sum j \in N(i) \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} (\mathbf{v}_i - \mathbf{v}_j)$.

To assess the smoothness of the generated mesh, we uniformly sampled 50 points within the range $[-1, 1]$ for each latent dimension. This approach resulted in 400 meshes for each method being evaluated. Subsequently, we calculated the mean loss for every mesh and vertex, and the results are recorded in Table 3. Our method demonstrates superior performance, closely aligning with



(a) Ours (b) VAE (c) β-VAE (d) β-TCVAE (e) PCA-FA

Figure 4: Disentanglement: For each method, we sample 50 points within a -3 to 3 range for each latent dimension and plot the corresponding changes in a feature. These features, listed from top to bottom, are height, arm length, crotch height, chest circumference, hip circumference, and waist circumference. The results clearly show that our method is highly effective, each feature is affected by only one latent variable, (giving interpretability and control), and unaffected by others (strong disentanglement). Other methods fare much worse. PCA-FA stands for PCA with feature analysis [Allen et al. 2003].

Statistics	Attributes					
	Height	Chest Circumference	Waist Circumference	Hip Circumference	Crotch Height	Arm Length
Test Loss (mm)	3.96	3.10	7.73	2.90	1.09	2.81
mean (mm)	1716.44	996.75	848.01	1050.17	773.54	612.61
std (mm)	107.99	124.10	144.34	113.03	55.73	45.99
median (mm)	1714.50	978.00	832.00	1031.00	771.00	612.00

Table 1: Test loss for the feature function network for various features, along with the dataset's attribute statistics for reference.

the template results. This observation is further confirmed by visual inspection in Figure 5.

In Figure 5, we generate the mean mesh for each method using latent variables set to zero. We calculate the smoothness loss for each mean mesh and also compute the difference in smoothness loss between each mean mesh and the template mesh. Both the statistical and visual analysis confirm that incorporating Smoothness Loss enables our method to closely match the smoothness of the template

Reconstruction Error (mm)	Model Type			
	Ours	VAE	β-VAE	β-TCVAE
mean	18.43	18.52	24.05	18.49
std	10.31	10.27	14.09	10.26
median	16.51	16.63	21.25	16.61

Table 2: Comparison of reconstruction error across various methods, measured in millimeters. The β-VAE method employs β-annealing to enhance disentanglement [Burgess et al. 2018].

mesh, thereby mitigating high-frequency noise in the generated meshes.

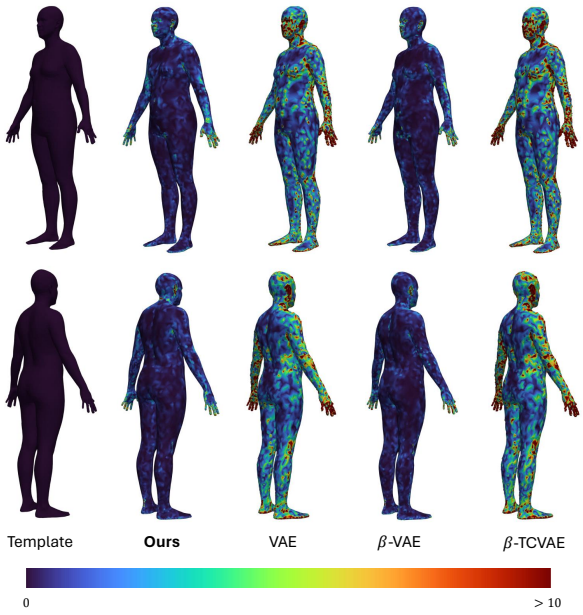


Figure 5: Comparison of the smoothness across various methods, with all meshes generated based on the mean shape. We plot the difference in Laplacian values between the template and those produced by each method. The template mesh used is the mean mesh derived from the dataset.

Smoothness	Model Type				
	Template	Ours	VAE	β-VAE	β-TCVAE
mean	12.20	12.64	16.86	13.23	17.74
std	28.45	28.41	25.43	30.15	24.70
median	4.10	4.16	6.77	4.20	7.79
max	1121.99	795.04	564.34	832.44	566.72

Table 3: Comparison of smoothness across different methods. We also use the mean mesh as a reference.

5.3 Disentanglement Quality

Our method is highly effective in disentangling selected features from the latent space. See Figures 4 and 6. In our experimental setup, six controlled features were chosen: height, waist circumference, chest circumference, hip circumference, arm length, and crotch height. Our Variational Autoencoder (VAE) has an 8-dimensional latent variable, with the first six dimensions representing these features respectively. Figure 6 shows a case of extreme disentanglement, where all dimensions are fixed except for one (waist circumference).

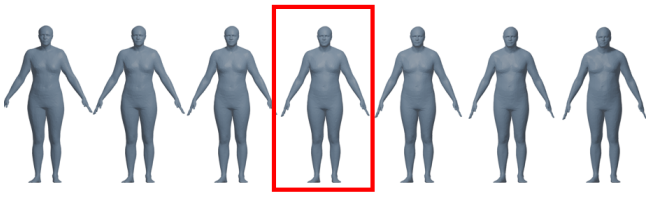


Figure 6: This showcases extreme disentanglement, where we maintained constant five latent dimensions influencing height, chest, hip, arm length, and crotch height, while varying the latent dimension associated with waist circumference.

In addition to a visual comparison, Figure 4 compares our method quantitatively with others to highlight its superiority. A distinct advantage of our method, as evident in Figure 4, is its unique capability to alter one feature independently, without impacting others. This demonstrates a level of disentanglement not achieved by other methods.

Note that some features, such as waist and hip circumference, are correlated. When we hold one constant while changing the other, the range of disentanglement is limited. If we try to extend the model beyond this limited range, it may produce abnormal shapes since there are no acceptable shapes in the space of human shapes.

6 ABLATION STUDY

Our method incorporates three primary components: disentanglement loss; smoothness; and techniques from multitask learning, which automatically balance the weights among the terms in our loss function. This section presents an ablation study to investigate the contributions of these components.

6.1 Disentanglement loss

Figure 7 compares the disentanglement behavior as various losses are removed. Figure 7b shows the crucial contribution of the disentanglement loss. When it is removed features are no longer disentangled, as evident in the slopes of the response curves. Other ablations do not significantly affect the disentanglement behavior, as shown in the other figures.

6.2 Smoothness loss

The quantitative results in Table 4 demonstrate that the absence of smoothness loss leads to an increase in smoothness error, evident in both mean and median error values. Figure 8 visualizes this difference over the body mesh.

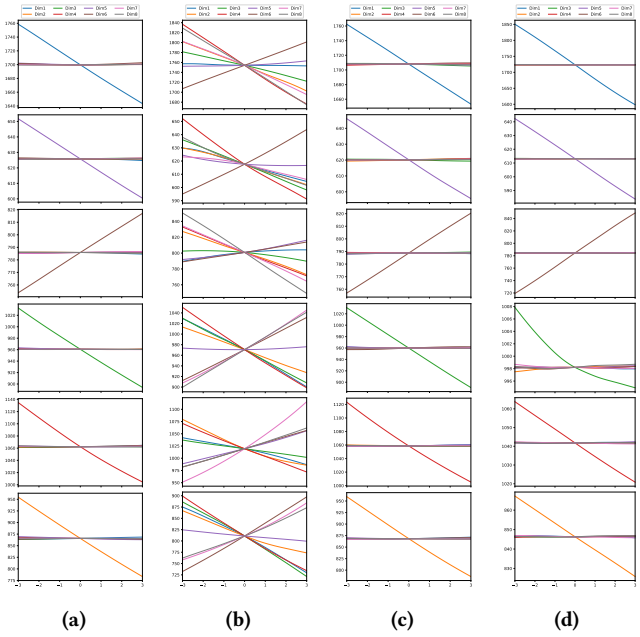


Figure 7: The features change in response to variations in the latent variable, ranging from -3 to 3, across each latent dimension for different methods. (a): Our method. (b): Our method without disentanglement loss. (c): Our method without Laplacian loss. (d): Our method without automatic weight balancing.

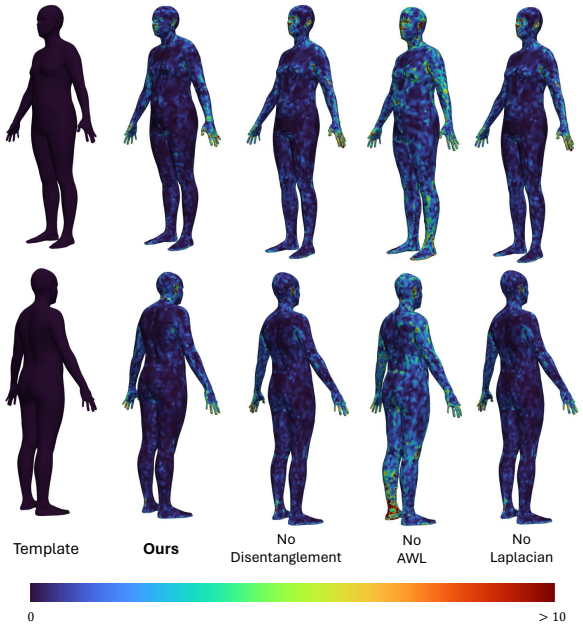


Figure 8: Ablation study: Comparison of the smoothness across various ablations, with all meshes generated based on the mean shape. We plot the difference in Laplacian values between the template and those produced by each method. The template mesh used is the mean mesh derived from the dataset.

Smoothness	Model Type				
	Template	Ours	No disentanglement	No smoothness	No AWL
mean	12.20	12.64	12.67	13.42	12.57
std	28.45	28.41	27.74	30.07	23.09
median	4.10	4.16	4.21	4.23	4.78
max	1121.99	795.04	672.22	887.25	397.04

Table 4: Comparison smoothness across different settings. We also use the mean mesh as a reference. No disentanglement: without disentanglement loss. No smoothness: without smoothness loss. No AWL: without using automatic weighted Loss.

Reconstruction Error (mm)	Model Type			
	Ours	No disentanglement	No Laplacian	No AWL
mean	18.43	18.13	18.53	22.80
std	10.31	10.07	10.41	13.44
median	16.51	16.27	16.59	20.13

Table 5: Comparison of mean reconstruction error across various settings, measured in millimeters. No dientanglement: without disentanglement loss. No Laplacian: without Laplacian loss. No AWL: without using automatic weighted Loss.

6.3 Automatic weight-balancing

We employ the weight-balancing techniques in multitask learning to automatically balance the various terms in our loss function, the effects of which are evident in multiple aspects. Table 5 shows that the absence of automatic weight balancing significantly increases reconstruction loss compared to other settings. Furthermore, Figure 7d indicates that the use of automatic weight balancing enhances the disentanglement results.

7 CONCLUSION AND LIMITATIONS

We presented RightSizing, a novel method for disentangling the latent space of 3D meshes. A significant aspect of our method is strong disentanglement of specific, measurable, features that are defined by the user. Strong disentanglement provides clear semantics for latent variables and enables precise control of targeted features. We learn a feature function for predicting 3D mesh features and use it with a generative model to predict not only the desired meshes but also their features and feature gradients. These feature gradients are a key part of our approach strong disentanglement. Experimental results demonstrate that RightSizing is highly effective and achieves significantly better disentanglement than recent methods without losing reconstruction quality. We demonstrated excellent results with human body meshes using a VAE model.

Our methodology has certain limitations. First, it requires continuous and differentiable features to disentangle. In particular, it falls short in disentangling categorical features such as gender. Second, in our current implementation, we learned the feature function

from human-measured training data. Such data may be expensive or impossible to obtain for some features. In the absence of such data it may be possible to design a suitable feature function that can be explicitly computed from mesh data. For example, we can use the maximum difference in the vertical coordinates as the height function and the mesh volume as a weight function. Third, in highly constrained cases where most of the major human body dimensions are fixed, unnatural shapes may appear with large changes in the remaining controlled variables because the model is forced to generate impossible shapes. Finally, our method has only been tested with a VAE model. However, our method only requires access to differentiable features, and could potentially be used to disentangle generative models other than VAEs.

ACKNOWLEDGEMENT

This work was supported in part by the NRC AI4D program, and by an NSERC Discovery grant to DKP.

REFERENCES

Mohammad Amin Aliari, Andre Beauchamp, Tiberiu Popa, and Eric Paquette. 2023. Face Editing Using Part-Based Optimization of the Latent Space. In *Computer Graphics Forum*, Vol. 42. Wiley Online Library, 269–279.

Brett Allen, Brian Curless, and Zoran Popović. 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)* 22, 3 (2003), 587–594.

Brian Amberg, Reinhard Knothe, and Thomas Vetter. 2008. Expression invariant 3D face recognition with a morphable model. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, 1–6.

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*. 408–416.

Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. 2019. Geometric Disentanglement for Generative Latent Shape Models. In *International Conference on Computer Vision, ICCV 2019*.

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

V Blanz and T Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1999)*. ACM Press, 187–194.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. 2016. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *Computer Vision – ECCV 2016 (Lecture Notes in Computer Science)*. Springer International Publishing.

Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. 2019. Neural 3D Morphable Models: Spiral Convolutional Networks for 3D Shape Representation Learning and Generation. In *The IEEE International Conference on Computer Vision (ICCV)*.

Leyde Briceno and Gunther Paul. 2019. MakeHuman: A Review of the Modelling Framework. In *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*, Sebastiano Bagnara, Riccardo Tartaglia, Sara Albolino, Thomas Alexander, and Yushi Fujita (Eds.). Springer International Publishing, Cham, 224–232.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599* (2018).

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems* 31 (2018).

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and P. Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Neural Information Processing Systems*.

Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. 2020. LIMP: Learning Latent Shape Representations with Metric Preservation Priors. In *Computer Vision – ECCV 2020*. Springer International Publishing, 19–35.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016).

Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. 2020. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5154–5163.

Cian Eastwood and Christopher KI Williams. 2018. A framework for the quantitative evaluation of disentangled representations. In *International conference on learning representations*.

Simone Foti, Bongjin Koo, Danail Stoyanov, and Matthew J. Clarkson. 2023. 3D Generative Model Latent Disentanglement via Local Eigenprojection. *Computer Graphics Forum* 42 (2023).

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

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

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. Meshcnn: a network with an edge. *ACM Transactions on Graphics (ToG)* 38, 4 (2019), 1–12.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. β-VAE: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*.

David A Hirshberg, Matthew Loper, Eric Rachlin, and Michael J Black. 2012. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*. Springer, 242–255.

Boyi Jiang, Juyong Zhang, Jianfei Cai, and Jianmin Zheng. 2019. Disentangled Human Body Embedding Based on Deep Hierarchical Neural Network. *IEEE Transactions on Visualization and Computer Graphics* 26 (2019), 2560–2575.

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-end Recovery of Human Shape and Pose. In *Computer Vision and Pattern Recognition (CVPR)*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7482–7491.

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

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Tejas D. Kulkarni, William F. Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum. 2015. Deep Convolutional Inverse Graphics Network. In *Neural Information Processing Systems*.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. 2018. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. In *International Conference on Learning Representations*.

Yann LeCun, Fu Jie Huang, and Leon Bottou. 2004. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 2. IEEE, II–104.

Lukas Liebel and Marco Körner. 2018. Auxiliary tasks in multi-task learning. *arXiv preprint arXiv:1805.06334* (2018).

Francesco Locatello. 2021. Enforcing and Discovering Structure in Machine Learning. *arXiv preprint arXiv:2111.13693* (2021).

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*. PMLR, 4114–4124.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

MakeHuman. 2024. MakeHuman, opensource tool for making 3D characters. http://www.makehumancommunity.org/.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. 2017. dSprites: Disentanglement testing Sprites dataset. https://github.com/deepmind/dsprites-dataset/.

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*. Springer, 35–57.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 10975–10985.

Ulrich Pinkall and Konrad Polthier. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* 2, 1 (1993), 15–36.

Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. 2018. Generating 3D faces using convolutional mesh autoencoders. In *European conference on computer vision (ECCV)*. 704–720.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. 2015. Deep visual analogy-making. *Advances in neural information processing systems* 28 (2015).

Karl Ridgeway and Michael C Mozer. 2018. Learning deep disentangled embeddings with the F-statistic loss. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 185–194.

Kathleen M Robinette, Sherri Blackwell, Hein Daanen, Mark Boehmer, Scott Fleming, Tina Brill, David Hoeferlin, and Dennis Burnsides. 2002. Civilian American and European surface anthropometry resource (CAESAR), final report, volume I: Summary. *Sytronics Inc Dayton Oh* (2002).

Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017).

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.

Olga Sorkine. 2005. Laplacian mesh processing. *Eurographics (State of the Art Reports)* 4, 4 (2005).

Stephan Streuber, M Alejandra Quiros-Ramirez, Matthew Q Hill, Carina A Hahn, Silvia Zuffi, Alice O'Toole, and Michael J Black. 2016. Body talk: Crowdshaping realistic 3D avatars with words. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–14.

Xiaokun Sun, Qiao Feng, Xiongzheng Li, Jinsong Zhang, Yu-Kun Lai, Jingyu Yang, and Kun Li. 2023. Learning Semantic-Aware Disentangled Representation for Flexible 3D Human Body Editing. In *CVPR 2023*.

Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

2387–2395.

Pengcheng Xi, Won-Sook Lee, and Chang Shu. 2007. Analysis of segmented human body scans. In *Proceedings of graphics interface 2007*. 19–26.

Fei Yang, Jue Wang, Eli Shechtman, Lubomir Bourdev, and Dimitri Metaxas. 2011. Expression flow for 3D-aware face component transfer. In *ACM SIGGRAPH 2011 papers*. 1–10.

Haitao Yang, Bo Sun, Liyan Chen, Amy Pavel, and Qixing Huang. 2023b. GeoLatent: A Geometric Approach to Latent Space Design for Deformable Shape Generators. *ACM Trans. Graph.* 42, 6 (2023), 242:1–242:20.

Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J. Guibas, and Lin Gao. 2023a. DSG-Net: Learning Disentangled Structure and Geometry for 3D Shape Generation. *ACM Trans. Graph.* 42, 1 (2023), 1:1–1:17.

Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. 2020a. Unsupervised Shape and Pose Disentanglement for 3D Meshes. In *European Conference on Computer Vision (ECCV)*.

Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. 2020b. Fully convolutional mesh autoencoder using efficient spatially varying kernels. *Advances in neural information processing systems* 33 (2020), 9251–9262.