UNSUPERVISED GEOMETRIC DISENTANGLEMENT FOR SURFACES VIA CFAN-VAE

Anonymous authors
Paper under double-blind review

ABSTRACT

Geometric disentanglement, the separation of latent codes for intrinsic (i.e. identity) and extrinsic (i.e. pose) geometry, is a prominent task for generative models of non-Euclidean data such as 3D deformable models. It provides greater interpretability of the latent space, and leads to more control in generation. This work introduces a mesh feature, the conformal factor and normal feature (CFAN), for use in mesh convolutional autoencoders. We further propose CFAN-VAE, a novel architecture that disentangles identity and pose using the CFAN feature. Requiring no label information on the identity or pose during training, CFAN-VAE achieves geometric disentanglement in an unsupervised way. Our comprehensive experiments, including reconstruction, interpolation, generation, and identity/pose transfer, demonstrate CFAN-VAE achieves state-of-the-art performance on unsupervised geometric disentanglement.

1 INTRODUCTION

With the recent popularity of geometric deep learning (Bronstein et al., 2017), mesh based convolutional autoencoders (MeshVAEs) are now a popular tool for surface generation in the surface processing community (Cheng et al., 2019; Litany et al., 2018; Ma et al., 2019; Ranjan et al., 2018). With these VAEs achieving state-of-the-art performance on tasks such as reconstruction, more attention is being given towards tasks such as latent space interpretability. Geometric disentanglement, where the latent variables controlling *intrinsic* (properties independent of surface embedding) and *extrinsic* (properties dependent on surface embedding) geometry are separated (Aumentado-Armstrong et al., 2019), is an important open problem related to such interpretability. A typical application of a disentangled latent space is the separation of identity and pose in the case of human body generation (Jiang et al., 2019a; Tan et al., 2018).

We are interested in creating an architecture that explicitly leads to geometric disentanglement in an unsupervised setting. Namely, we do not require labels for mesh identity and pose, allowing the architecture to be applied more broadly. In this work, we:

1. introduce a feature, the conformal factor and normal feature (CFAN), that decouples intrinsic and extrinsic geometry for use in mesh convolutional autoencoders.
2. propose a novel architecture, CFAN-VAE, for unsupervised geometric disentanglement. For a given mesh, we compute the CFAN feature, and encode its components separately into latent vectors representing intrinsic and extrinsic geometry.

Related Work Geometric disentanglement is an important feature leading to a more interpretable latent space and thus more control in generation. For instance, (Jiang et al., 2019a;b; Tan et al., 2018) learn a latent space separating identity and pose, advocating for the use of the *as-consistent-as-possible deformation representation* (ACAPDR). These works require a reference pose for all identities during training. Recently, several works proposed networks using non-Euclidean convolution for this task (Cosmo et al., 2020; Levinson et al., 2019; Zhou et al., 2020). These networks either utilize supervision or loss functions that require identification of isometric pairs in training data. In practice, the latter needs supervised identity to scale on large datasets. Training datasets lacking isometries may be desireable in applications such as differential privacy (Abadi et al., 2016).

There is limited work to achieve unsupervised geometric disentanglement that *completely* eliminates reliance on both identity and pose labeling. To our knowledge, the only previous work concerning

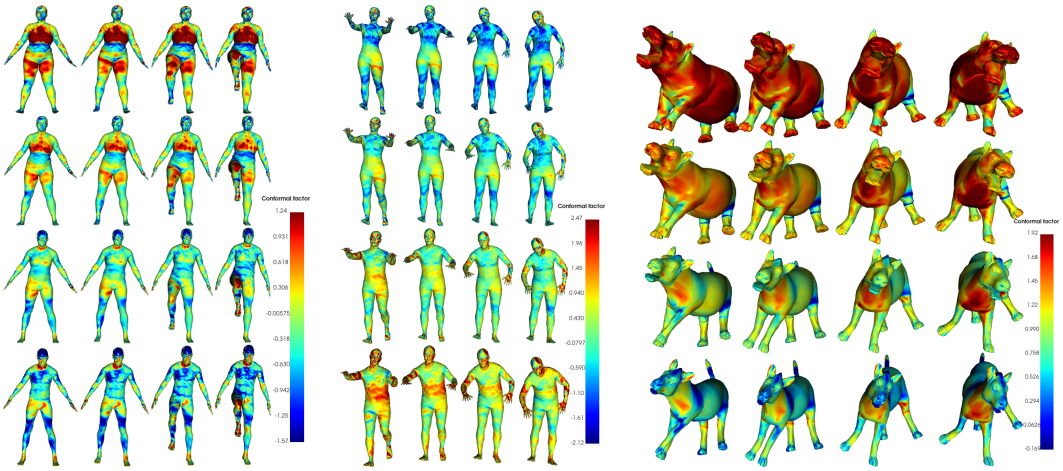

Figure 1: Geometric disentangled interpolations between two meshes from DFAUST, SURREAL, and SMAL datasets. Meshes are generated by CFAN-VAEs. The horizontal/vertical axis display linear interpolation in the normal/conformal latent codes, $\boldsymbol{z}_n/\boldsymbol{z}_c$. Color denotes the conformal factor feature of each reconstructed mesh.

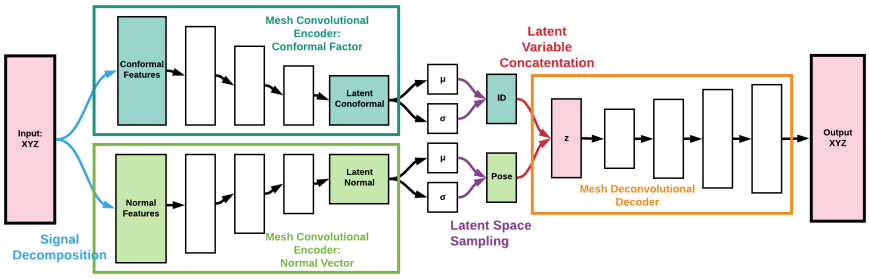

Figure 2: Diagram of the CFAN-VAE architecture. 3D mesh coordinates are transformed to the CFAN feature and separately encoded to promote disentanglement.

unsupervised disentanglement for 3D data is (Aumentado-Armstrong et al., 2019), in which the authors introduce GDVAE, a VAE built on a PointNet (Charles et al., 2017).

2 CFAN-VAE

Our novel method for geometric disentanglement is motivated by the Fundamental Theorem of Surfaces (Do Carmo, 2016). We describe surfaces using conformal factors and surface normal vectors. This CFAN feature is easily leveraged by our introduced CFAN-VAE for geometric disentanglement.

CFAN Feature The Fundamental Theorem of Surfaces states that a surface can be uniquely reconstructed up to rigid motion given feasible metric tensor and surface normals (Do Carmo, 2016). Then we can use metric tensors and normal vector fields to characterize surfaces. In this work, we consider genus zero surfaces. It is well known that all genus zero surfaces are conformally equivalent (Jost & Jost, 2008). Namely, given two genus-zero surfaces, $(\mathcal{M}_1, g_1)$ and $(\mathcal{M}_2, g_2)$, there exists a diffeomorphism, $\phi : (\mathcal{M}_1, g_1) \to (\mathcal{M}_2, g_2)$ where $\phi^*(g_2) = \exp(2\lambda) g_1$.

Here the function, λ, is known as the conformal factor and defines a conformal deformation from $\mathcal{M}_1$ to $\mathcal{M}_2$. Then any pair of conformally equivalent surfaces can be deformed into one another (up to isomorphism) by choosing the correct conformal factor. The surface embedding is defined up to translation by the normal field. These two components define the CFAN feature, with the conformal factor and normal field representing intrinsic and extrinsic geometry respectively.

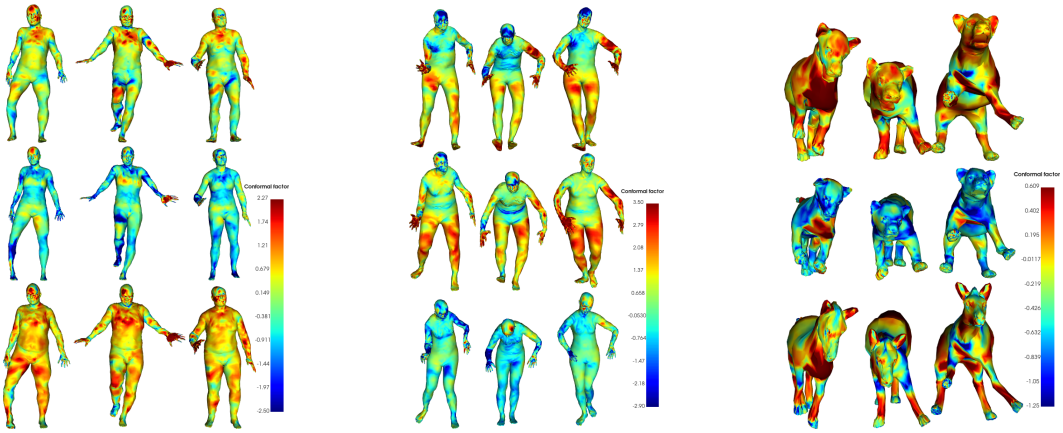

Figure 3: Surfaces generated by sampling the CFAN-VAE latent space. Horizontal and vertical axes correspond to $\boldsymbol{z}_n$ and $\boldsymbol{z}_c$ respectively. CFAN-VAE reliably generates meshes in a specific pose or with nearly the same identity, showing geometric disentanglement. Color denotes conformal factor.

Given a triangle mesh $(\boldsymbol{P}, \boldsymbol{T})$, where $\boldsymbol{P} \in \mathbb{R}^{n\times 3}$ is the set of vertices and $\boldsymbol{T} \in \mathbb{R}^{k\times 3}$ is the corresponding set of faces, we define the discretized CFAN feature. Each face τ in $\boldsymbol{T}$ has a corresponding exterior face normal $\boldsymbol{n}_\tau$. We compute the weighted average of these face normals around the first ring structure of each vertex to define a pointwise normal. Since the local area on the surface is given by $\sqrt{\det g}$, it follows that the logarithm of local area is proportional to the conformal factor. Then we define the CFAN feature, $(c_i, \boldsymbol{n}_i)$ as:

$$c_i = \log \sum_{\substack{\tau\in\boldsymbol{T};\\ i\in\tau}} \frac{\text{Area}(\tau)}{3}, \qquad \boldsymbol{n}_i = \frac{\sum_{\substack{\tau\in\boldsymbol{T};\\ i\in\tau}} \text{Area}(\tau)\boldsymbol{n}_\tau}{||\sum_{\substack{\tau\in\boldsymbol{T};\\ i\in\tau}} \text{Area}(\tau)||} \tag{1}$$

Network Architecture Based on the CFAN feature, we propose a simple architecture, CFAN-VAE, as shown in Figure 2 to achieve unsupervised geometric disentanglement. The intuition is to encode the conformal factor and the normal features separately. The 3D coordinates $\boldsymbol{p}$ of an input mesh under a fixed triangulation are first converted to the CFAN feature by transformations ϕ_c and ϕ_n provided by equation 1, yielding $\boldsymbol{c} = \phi_c(\boldsymbol{p})$ and $\boldsymbol{n} = \phi_n(\boldsymbol{p})$.

Then, the feature components are separately encoded by E_c and E_n to create two different latent variables, the conformal latent variable $\boldsymbol{z}_c$ and the normal latent variable $\boldsymbol{z}_n$ in the disentangled latent space $\boldsymbol{Z}_{c,n} = \boldsymbol{Z}_c \times \boldsymbol{Z}_n$. That is, $\boldsymbol{z}_c = E_c(\boldsymbol{c})$, $\boldsymbol{z}_n = E_n(\boldsymbol{n})$, and $\hat{\boldsymbol{p}} = D(\boldsymbol{z}_c, \boldsymbol{z}_n)$.

As a result, $\boldsymbol{z}_c$ corresponds to intrinsic geometry, controlling surface identity, and $\boldsymbol{z}_n$ corresponds to extrinsic geometry, controlling surface pose. After that, the CFAN latent variable $\boldsymbol{z}_{c,n} = [\boldsymbol{z}_c, \boldsymbol{z}_n] \in \boldsymbol{Z}_{c,n}$ is decoded by D to obtain the reconstructed 3D coordinates $\hat{\boldsymbol{p}}$,

Our model is built on geometric convolutions given by PTC layers (Schonsheck et al., 2018). The mesh signal is decimated during encoding and refined during decoding by a factor of four. Sampling is precomputed by the geodesic Farthest Point Sampling (FPS) method (Moenning & Dodgson, 2003). After the final mesh convolution in the encoder, dense layers map the signal to the variational statistics, $\mu := [\mu_c, \mu_n]$ and $\sigma := [\sigma_c, \sigma_n]$. A dense layer also maps $\boldsymbol{z}_{c,n}$ to the first mesh signal in the decoder. Details on the loss function used for training can be found in Section A.1.

3 Numerical Experiments and Results

We consider three datasets in our experiments. This includes DFAUST, a real-world motion-capture dataset of 10 people performing several poses (Bogo et al., 2017). We randomly split the dataset into a training/validation/test set of 35,720/500/5,000 meshes respectively. In addition, we generated a synthetic dataset of animals referred to as SMAL following (Zuffi et al., 2017), based on the SMPL model introduced in (Loper et al., 2015). Similary, we formed a synthetic dataset of

Table 1: Chamfer distance of the mesh generated in pose and identity transfer tasks to the target mesh. The generated mesh is decoded from swapping the latent codes of the source meshes, $(\boldsymbol{z}_c^{(1)}, \boldsymbol{z}_n^{(1)})$ and $(\boldsymbol{z}_c^{(2)}, \boldsymbol{z}_n^{(2)})$. This distance is the mean over 256 transfer pairs with standard error.

	SURREAL		SMAL	
Networks	Error(mm)	#Param.(M)	Error(mm)	#Param.(M)
Distance between sources	76.0 ± 1.8	—	313.9 ± 9.7	—
GDVAE [1]	31.6 ± 0.6	21.30	96.3 ± 1.9	21.30
CFAN-VAE	$\mathbf{26.8} \pm \mathbf{0.8}$	1.75	$\mathbf{35.5} \pm \mathbf{0.8}$	2.15

humans referred to as SURREAL from (Varol et al., 2017). These datasets are created following (Aumentado-Armstrong et al., 2019). For SURREAL, we add small normal noise with $\sigma = 0.2$ to the intrinsic shape parameters β to eliminate any isometry in the dataset. These two datasets provide challenging environments where identity-supervised methods fail.

Details on the network and training hyperparameters are detailed in Section A.2. For comparisons to GDVAE, we use the available dataset-specific pretrained models available in (Aumentado-Armstrong et al., 2019). We verify that CFAN-VAE is able to accurately reconstruct surfaces. Figure 4 displays high quality reconstructions for each dataset using CFAN-VAEs for DFAUST, SURREAL, and SMAL.

Surface Interpolation Figure 1 displays examples of disentangled interpolations for each dataset using CFAN-VAE. In this figure, the vertical and horizontal axis represents conformal and normal interpolations respectively. The color map denotes the pointwise normalized conformal factor of the reconstructions. These results verify that CFAN-VAE produces meaningful geometric disentanglement between intrinsic and extrinsic information. Thus, it enjoys flexibility to generate high quality meshes preserving either identity or pose. Notice, the colormaps and position are almost identical for fixed $\boldsymbol{z}_c$ and $\boldsymbol{z}_n$ respectively. We include additional examples in Figure 5 in the appendix.

We compare the performance of CFAN-VAE to GDVAE on transfer tasks. For SURREAL and SMAL, we can create ground-truth mesh solutions to the transfer task by exchanging the appropriate SMPL parameters of the source meshes. Then we compare the Chamfer distance of the network generated mesh, with appropriate latent code exchanged, to the SMPL generated ground truth. These results are summarized in Table 1. It is clear that CFAN-VAE outperforms GDVAE, particularly on the SMAL dataset. This nice property of the CFAN-VAE latent space enables flexible control of the geometry of generated surfaces. We stress that these two datasets lack isometric pairs, so other methods (Cosmo et al., 2020; Zhou et al., 2020) are not of use for comparison. Further analysis of the learned latent space is available in Section A.3.

Surface Generation Figure 3 plots random generations where latent codes are sampled from a multivariate Gaussian. This Gaussian, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, is a product of marginals fit to the embedding of the test set for each latent variable. We sample from $\mathcal{N}(\boldsymbol{\mu}, 0.8\boldsymbol{\Sigma})$. Here the normal latent code and conformal code is fixed along the vertical axis and horizontal axis respectively. Clearly, pose and identity are successfully fixed. The generated meshes are of comparable visual quality to the reconstructions from the test set. Then the latent space of CFAE-VAE allows for high-quality generation while providing interpretability of geometric features.

4 Conclusion

We propose a novel architecture, CFAN-VAE, for unsupervised geometric disentanglement in mesh convolutional autoencoders. This is accomplished by utilizing the conformal factor and normal (CFAN) feature to separate intrinsic and extrinsic geometry. There is clearly strong geometric disentanglement in CFAN-VAEs. We achieve state-of-the-art performance on transfer tasks and disentanglement compared to prior work. The continued integration of geometric theory into neural network architectures can only lead to more interesting results in the future.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. Geometric disentanglement for generative latent shape models, 2019.

Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, Jul 2017. ISSN 1053-5888. doi: 10.1109/msp.2017.2693418. URL http://dx.doi.org/10.1109/MSP.2017.2693418.

R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. doi: 10.1109/cvpr.2017.16. URL http://dx.doi.org/10.1109/cvpr.2017.16.

Shiyang Cheng, Michael Bronstein, Yuxiang Zhou, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. Meshgan: Non-linear 3d morphable models of faces, 2019.

Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. Limp: Learning latent shape representations with metric preservation priors, 2020.

Manfredo P Do Carmo. *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications, 2016.

Boyi Jiang, Juyong Zhang, Jianfei Cai, and Jianmin Zheng. Learning 3d human body embedding, 2019a.

Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. Disentangled representation learning for 3d face shape, 2019b.

Jürgen Jost and Jèurgen Jost. *Riemannian geometry and geometric analysis*, volume 42005. Springer, 2008.

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

Jake Levinson, Avneesh Sud, and Ameesh Makadia. Latent feature disentanglement for 3d meshes, 2019.

Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. doi: 10.1109/cvpr.2018.00202. URL http://dx.doi.org/10.1109/CVPR.2018.00202.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34:248:1–248:16, 2015.

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

Qianli Ma, Siyu Tang, Sergi Pujades, Gerard Pons-Moll, Anurag Ranjan, and Michael J. Black. Dressing 3d humans using a conditional mesh-vae-gan. *ArXiv*, abs/1907.13615, 2019.

Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003.

Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3d faces using convolutional mesh autoencoders. *Lecture Notes in Computer Science*, pp. 725–741, 2018. ISSN 1611-3349.

Stefan C. Schonsheck, Bin Dong, and Rongjie Lai. Parallel transport convolution: A new tool for convolutional neural networks on manifolds, 2018.

Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. Variational autoencoders for deforming 3d mesh models. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. doi: 10.1109/cvpr.2018.00612. URL http://dx.doi.org/10.1109/cvpr.2018.00612.

Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Unsupervised shape and pose disentanglement for 3d meshes. In *16th European Conference on Computer Vision*. Springer, 2020.

Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

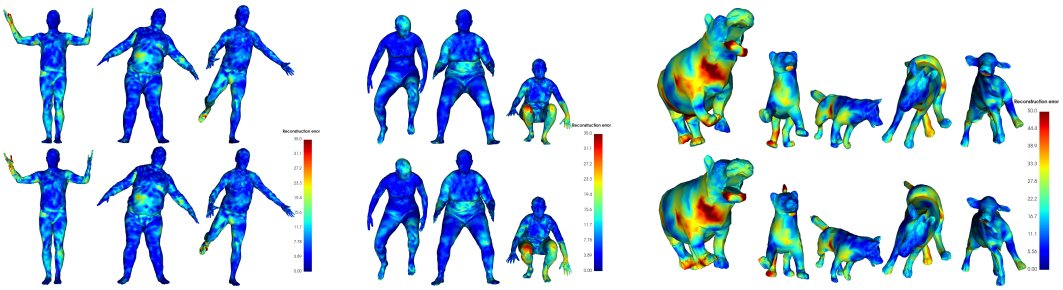

Figure 4: Reconstructions from the CFAN-VAE models in Figure 1. The mesh color represents the pointwise Euclidean error after translation. The top row are the ground truth meshes, and the bottom row are the reconstructions.

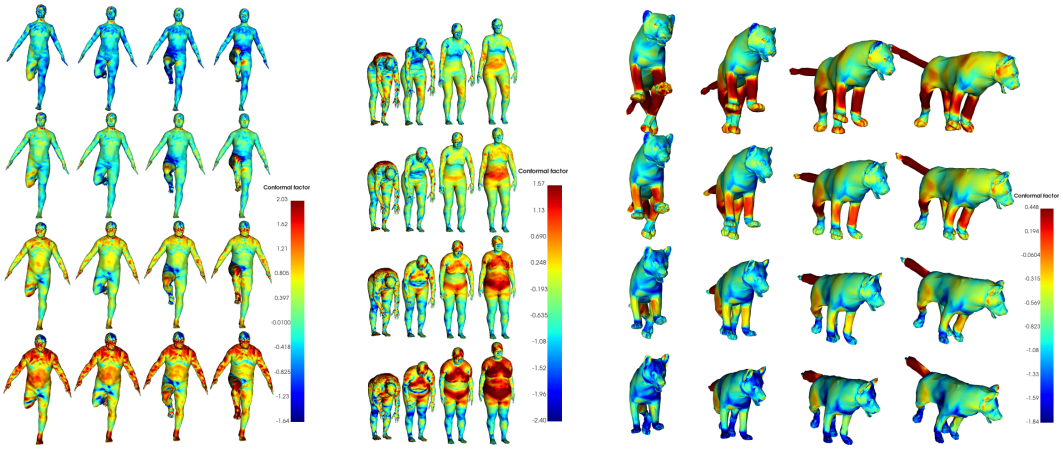

Figure 5: Additional geometrically disentangled interpolations using CFAN-VAE as in Figure 1.

A Appendix

A.1 Loss Function

The training loss for our model is given by:

$$\mathcal{L} := ||\boldsymbol{P} - \hat{\boldsymbol{P}}||_1 + \sum_{\mu^{(0)},\mu^{(1)} \in \mu} \Big(\lambda_D \mathcal{L}_D(\mu^{(0)}, \mu^{(1)}) + \lambda_M \mathcal{L}_M(\mu^{(0)}, \mu^{(1)})\Big) + \lambda_{KL}||\sigma^2 + \mu^2 - 1 - \log(\sigma^2)||_1. \tag{2}$$

The first term is simply the L_1 error of the reconstruction which promotes robustness to vertex outliers. $\mathcal{L}_D$ and $\mathcal{L}_M$ are disentanglement and metric penalties that we define shortly. The fourth term of the loss function is the KL-divergence of the latent representation from the unit normal distribution, given by the Bayesian prior assumption (Kingma & Welling, 2013).

In CFAN-VAE, geometric disentanglement is a result of separately encoding intrinsic and extrinsic geometric information. We stress that structural conditions prevent guaranteeing complete independence of the separate latent vectors. This is intuitive, considering motion can be restricted by intrinsic properties such as body mass distribution.

To promote independence, we consider a pair of encoded meshes. We create a latent variable with the conformal latent code of the first mesh, while the normal code is a random convex combination of the two normal codes. We decode and re-encode this new latent variable. We then have an L_2 penalty on the resulting change in the conformal code upon re-encoding as it should not be affected by this change in the normal code. We penalize a change in the normal code analogously. This

Table 2: Disentanglement metrics on the learned latent spaces. For each mesh embedding, we find its nearest neighbor in either the intrinsic or extrinsic space, $\boldsymbol{z}_c$ / $\boldsymbol{z}_n$. We report the norm of the difference in the identity and pose SMPL parameters, β and θ, of the embedded mesh and its neighbor. We report mean distance with standard error on the test set.

Networks	Param.	Desired Magnitude		SURREAL		SMAL	
				$\boldsymbol{z}_c$	$\boldsymbol{z}_n$	$\boldsymbol{z}_c$	$\boldsymbol{z}_n$
GDVAE [2]	β	$\downarrow$	$\uparrow$	4.42 ± 0.03	4.44 ± 0.03	2.40 ± 0.02	3.48 ± 0.02
	θ	$\uparrow$	$\downarrow$	4.21 ± 0.04	4.11 ± 0.04	2.72 ± 0.01	2.71 ± 0.01
CFAN-VAE	β			$\mathbf{4.05 \pm 0.02}$	4.38 ± 0.02	$\mathbf{1.73 \pm 0.01}$	3.73 ± 0.02
	θ			3.69 ± 0.03	$\mathbf{2.81 \pm 0.04}$	2.74 ± 0.01	$\mathbf{2.63 \pm 0.01}$

disentanglement penalty, $\mathcal{L}_{\mathcal{D}}$, is formalized below,

$$\mathcal{L}_D := \mathbb{E} \sum_{\substack{i,j \text{ in} \\ \{c,n\}}} ||\mu_i^{(0)} - E_i \left(D(\mu_c^{(0)} + \delta_{j,c}\epsilon_c, \mu_n^{(0)} + \delta_{j,n}\epsilon_n) \right) ||_2^2, \tag{3}$$

$$\epsilon_k := \alpha(\mu_k^{(1)} - \mu_k^{(0)}), \quad \alpha \in U[0,1]. \tag{4}$$

Inspired by the metric regularization introduced in (Cosmo et al., 2020), we add a metric penalty, $\mathcal{L}_M$, to the loss function,

$$\mathcal{L}_M := \mathbb{E}_\alpha ||l_\alpha(\phi_c \circ D(\mu^{(0)}), \phi_c \circ D(\mu^{(0)})) - \phi_c \circ D\left(l_\alpha(\mu^{(0)}, \mu^{(1)})\right) ||_1, \quad \alpha \in U[0,1], \tag{5}$$

where l_α is the α convex combination of its entries. This promotes smoothness in the metric of reconstructions.

A.2 Network Hyperparameters

Each VAE contains 5 layer encoders and decoder. For the PTC kernels, we use a 13 point stencil with the support of the kernel about a point being its 9 nearest geodesic neighbors. The size of the conformal/normal latent vectors are (32, 32) for DFAUST and (8, 32) for SMAL. For visualization, we use models with a latent dimension of (32, 32). For corresponding xyz-VAEs, the latent dimension is the sum of the two latent dimensions.

The latent dimension of the GDVAE models are $(5, 15)$ and $(5, 11)$ for SURREAL and SMAL respectively. When comparing to CFAN-VAE models for SURREAL, we choose a corresponding latent dimension of $(5, 15)$.

We use batch normalization (BN) layers in the encoder. When training with the disentanglement penalty $\mathcal{L}_D$, we freeze the BN layers during reencoding. The activation functions are ReLU and ELU for the encoders and decoder. We use the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 1E-3 and a weight decay of 5E-5. The value of λ_{KL} is set at 1E-4, while λ_D and λ_M are 5E-2. Each model is trained for 300 epochs with a batch size of 32. We train two instances of each network using different random seeds.

A.3 Latent Space Analysis

To better assess the geometric disentanglement in our networks, we analyze the latent spaces of the CFAN-VAEs. We quantitatively measure the disentanglement in latent space in the manner of (Zhou et al., 2020). First, we embed the test sets for SURREAL and SMAL in the latent space. Given a mesh embedding, we find its nearest neighbor in either the intrinsic or extrinsic latent spaces, $\boldsymbol{Z}_c$ or $\boldsymbol{Z}_n$. As these meshes were created via SMPL shape and pose parameters, β and θ, we measure the distance between these parameters of the embedded mesh and its neighbor. These distances are summarized in Table 2. If the network latent spaces truly possess disentanglement, we expect the $\boldsymbol{z}_c$ neighbor to be closer in β distance and the $\boldsymbol{z}_n$ neighbor to be closer in θ distance.