Supplementary Material: Dense Deformation Embedding Network for Template-Free Shape Correspondence

In this supplementary material, we first provide detailed architecture of the Extrinsic-Intrisic Autoencoder and the global embedding baseline, in Sec. 1. We then describe the details on construction of the mesh hierarchy in Sec 2. In Sec. 3, we provides more analysis on the effect of the Extrinsic-Intrisic Autoencoder. Sec. 4 provides more ablation studies on employed losses. We evaluate the model complexity in Sec. 5. Sec. 6 provides more details on the methods that we compare with in the experiments of the main paper. In Sec. 7, we show more qualitative results on mesh deformation. In Sec. 8, we provide more visualization results for applications of our model in human pose transfer, shape retrieval and shape interpolation.

1. Network Architecture

1.1. Extrinsic-Intrinsic Autoencoder

As shown in Figure 1, we present the detailed architecture of our proposed Extrinsic-Intrinsic Autoencoder (EI-AE). It consists of a two-layer encoder and a three-layer decoder. We apply *BatchNorm* and *ReLU* after each convolution layer, except for the bottleneck layer where the canonical shape $\mathcal{C}$ is directly output by a convolution layer. Without specification, we apply a 10D shared canonical shape $\mathcal{C}$ (*i.e.*, $e = 10$). We also apply 3D shared canonical shape in ablation study.

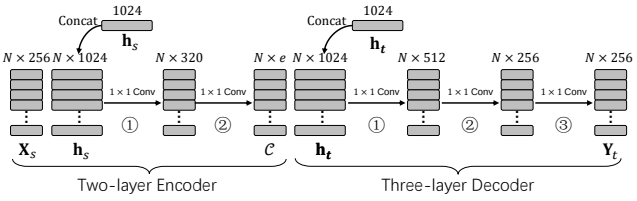

Figure 1. The architecture of the Extrinsic-Intrinsic Autoencoder.

1.2. Baseline

Figure 2 illustrates the architecture of our *global embedding* (GE) baseline. It employs the same Siamese mesh encoder E_g and deformation decoder D_d as our proposed UD^{2}E-Net. The only difference between them is that the GE baseline removes the EI-AE and directly concatenates source local features $\mathbf{X}_s$ with the target global feature $\mathbf{h}_t$ as deformation embeddings.

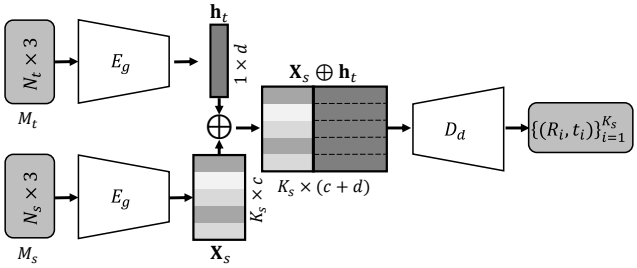

Figure 2. The architecture of the *global embedding* baseline.

2. Mesh Hierarchy

As shown in Figure 3, we utilize the Graclus algorithm [5] to construct the mesh hierarchy (M_1, M_2, M_3, M_4), where the coarsest mesh M_4 is the deformation graph $\mathcal{G}$ and M_1 is the input mesh. We construct mesh hierarchy for two main purposes. First, the meshes are downsampled to enable graph pooling on feature maps, which can not only reduce number of parameters and computational complexity by scaling down the size of feature maps but also enable *Graph Convolution Networks* (GCNs) to better aggregate information with enlarged receptive field sizes and learn hierarchical representations. Second, we downsample the input mesh to utilize the derived low-resolution mesh as the deformation graph $\mathcal{G}$.

Recently, DEMEA [15] also defines graph convolutions on a mesh hierarchy to learn deformation embeddings for a deformation graph. However, in [15], the mesh hierarchy is computed once prior to the training process using QEM based methods. This requires all training data to share the same topology, which vastly limits its application. We address this challenge by using the Graclus algorithm to downsample the graph in real-time simultaneously with the forwarding of the network, due to its high efficiency. Graclus traverses all nodes in the graph and in each step greedily merges two unmarked nodes that maxi-

mize the local normalized cut $w_{ij}(d_i^{-1} + d_j^{-1})$, then mark them as visited. When all nodes are visited, the graph has approximately half of the nodes. The graph hierarchy can be obtained by repeating this process until required resolution. Due to randomness of the algorithm, the derived mesh hierarchy can be updated dynamically, which also improves the generalization ability for GCNs.

Moreover, the vertex number of each mesh level is also not fixed. The vertex number of the input mesh is fixed as 2757, whereas the rest mesh levels have around 1480, 800, 430 vertices, respectively.

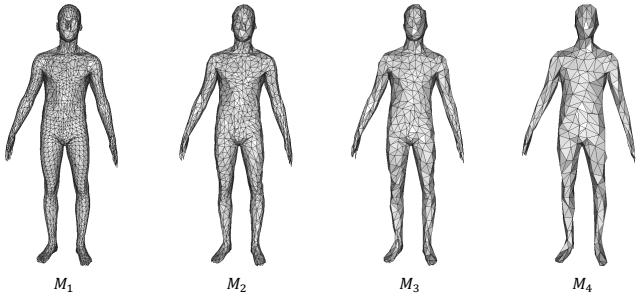

Figure 3. The mesh hierarchy generated by Graclus algorithm [5].

3. Effect of the Extrinsic-Intrinsic Autoencoder

In this section, we provide more visualization results for the proposed EI-AE. As shown in Figure 4, we visualize the learned 10D shared canonical shape $\mathcal{C}$ via t-SNE [16]. Similar to the 3D shared canonical shape, the 10D shared canonical shape also presents the shape of the skeleton of a body. However, without the guidance of the bounded *Maximum Mean Discrepancy* (MMD), the EI-AE fails to form a compact canonical shape, which demonstrates the indispensable role of the introduced bounded MMD.

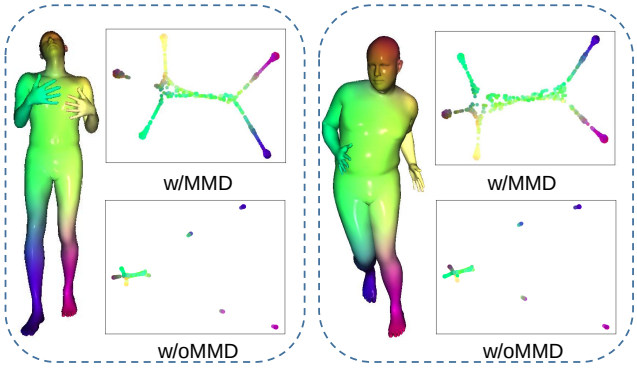

Figure 4. Visualization of 10D shared canonical shape $\mathcal{C}$ via t-SNE, where the left is the input shape and the top-right and bottom-right are the learned $\mathcal{C}$ with and without bounded MMD, respectively.

4. More ablation studies

4.1. Bounded Maximum Mean Discrepancy

We conduct ablation studies about β in Equation. 6 of the main paper among $\{0, 0.01, 0.1\}$ on SURREAL (230k) dataset. As shown in Table 1, the performance is better when $\beta > 0$, and is not quite sensitive to the choice of β. It proves that the hinge loss prevents the feature corruption, as illustrated in Section 3.3 of the main paper.

β	0	0.1	0.01
C_{DE} (cm)	2.30	2.04	**1.91**

Table 1. Ablation studies about β on SURREAL (230k) dataset.

4.2. Cycle-Consistent Loss

Both bounded MMD and the cycle-consistent loss can provide self-supervision for our UD^{2}E-Net. Here we discuss their effect on the construction of the canonical shape $\mathcal{C}$. We visualize the learned canonical shape in Figure 5. Specifically, our model without MMD fails to form a compact canonical shape. However, discarding the cycle-consistent loss does not degenerate the representation, which proves the indispensable role of MMD over the cycle-consistent loss.

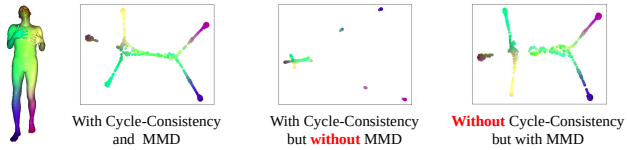

Figure 5. Representation visualization comparison with/without MMD and cycle-consistent loss.

5. Timing and model complexity

We measure the run-time of UD^{2}E-Net on Faust Dataset. It takes 68.32ms to process a pair of watertight meshes with 6890 vertices. The measurements are performed on an NVIDIA TITAN XP GPU across 500 runs. The entire network has 3.26M parameters.

6. Comparison Methods

The methods we compare with can be divided into deformation-based methods [7, 4, 9, 15], mesh autoencoders [13, 3] and spectral methods [10, 8, 6].

For deformation-based methods, **3D-CODED** [7] deforms a fixed template to align with the input target shape. It encodes the target shape globally into a 1024-D vector with a PointNet-like [12] encoder, then predicts a deformed location for each point on the template by decoding the concatenation of the global vector and the point's location. The method is fully supervised with ground-truth correspondences. It also has an unsupervised variant, which

we denote as Unsup. 3D-CODED in Table. 1,4 of the main paper. **Elementary** [4] extends 3D-CODED by automatically learning a better elementary structure from a shape collection for shape reconstruction and matching. It also deforms the input shape by predicting per-point locations and is fully supervised. In all experiments, we apply its 3D *patch deformation* variant, since it achieves the best performance on Faust benchmark [1]. DEMEA employs an embedded deformation layer based on [14] to deform a fixed template to restore the input shape, which is also fully supervised. **LBS-Autoencoder** [9] utilizes Linear Blending Skinning (LBS) for deformation, which is self-supervised with ground-truth joint rotation angles.

The mesh autoencoders [13, 3] encode the input mesh into a latent code with graph convolutions, and then directly decode it to restore the input mesh.

Spectral methods perform matching in the spectral domain. They are built upon a functional map representation [11] to learn descriptors for matching. **FMNet** [10] is supervised with ground-truth correspondences. **Halimi *et al.*** [8] assume isometric deformations and remove the supervision by minimizing pair-wise geodesic distance distortions. **Ginzburg *et al.*** [6] introduce cyclic mapping, which can generalize to non-isometric deformation and achieves state-of-the-art performance among unsupervised methods.

Our proposed UD^{2}E-Net outperforms above supervised and unsupervised methods on SURREAL and DFAUST dataset. On Faust benchmark [1], the proposed UD^{2}E-Net outperforms state-of-the-art unsupervised methods by 24%$\sim$37% on Inter challenge, and meanwhile achieves the best on Intra challenge even comparing to supervised methods.

7. Qualitative Results

In Figure 6, we show more qualitative results for deformation prediction on SURREAL 23k [17] dataset, where comparing methods suffer from both unrealistic artifacts and large reconstruction error due to large non-rigid deformations, whereas our proposed UD^{2}E-Net can yield more natural and accurate deformations.

8. Applications

In Figure 7, we show more human pose transfer results on DFAUST dataset [2], where the poses are successfully transferred from the source shape to the target shape. In Figure 8, we show more shape retrieval results on SURREAL 23k dataset [17], where the poses of the query and retrieved shapes are extremely similar.

Here, we further evaluate UD^{2}E-Net on shape interpolation task. Although UD^{2}E-Net does not follow a strict Autoencoder architecture, which is known to be good at forming a latent space, it turns out that UD^{2}E-Net still forms a surprisingly well-behaved latent space. Given two target meshes, we linearly interpolate their global features $\mathbf{h}_0, \mathbf{h}_1$ by $\mathbf{h}_{inter}(t) = (1-t)\mathbf{h}_0 + t\mathbf{h}_1$. As shown in Figure 9, $\mathbf{h}_{inter}$ can yield plausible in-between meshes in both figure and pose.

References

[1] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE. 3

[2] 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. 3

[3] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7213–7222, 2019. 2, 3

[4] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *Advances in Neural Information Processing Systems*, pages 7435–7445, 2019. 2, 3, 4

[5] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007. 1, 2

[6] Dvir Ginzburg and Dan Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *European Conference on Computer Vision*, pages 36–52. Springer, 2020. 2, 3

[7] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 2, 4

[8] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019. 2, 3

[9] Chun-Liang Li, Tomas Simon, Jason Saragih, Barnabás Póczos, and Yaser Sheikh. Lbs autoencoder: Self-supervised fitting of articulated meshes to point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11967–11976, 2019. 2, 3

[10] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017. 2, 3

[11] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible

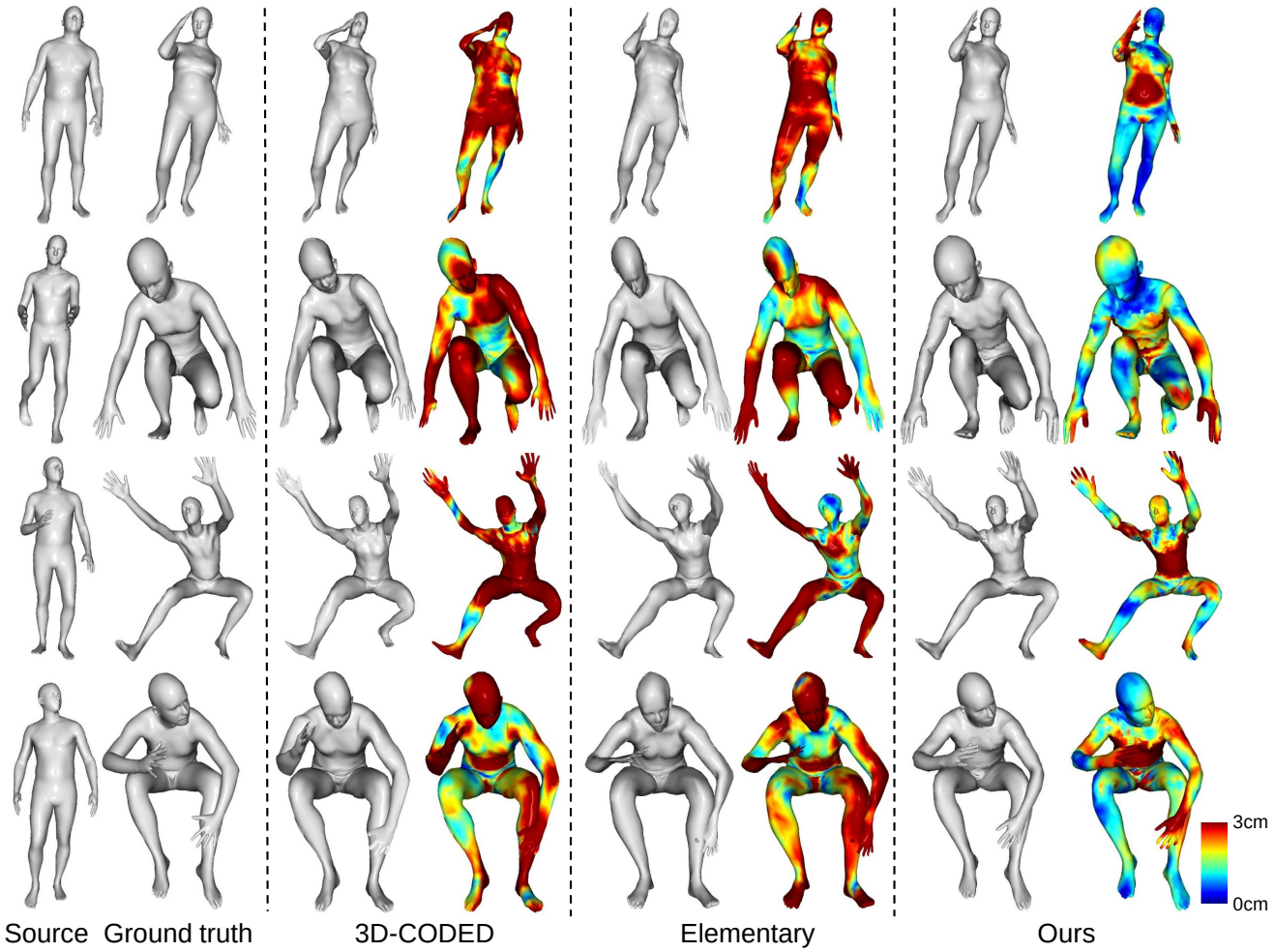

Figure 6. Qualitative comparison on SURREAL 23k [17] with 3D-CODED [7] and Elementary [4].

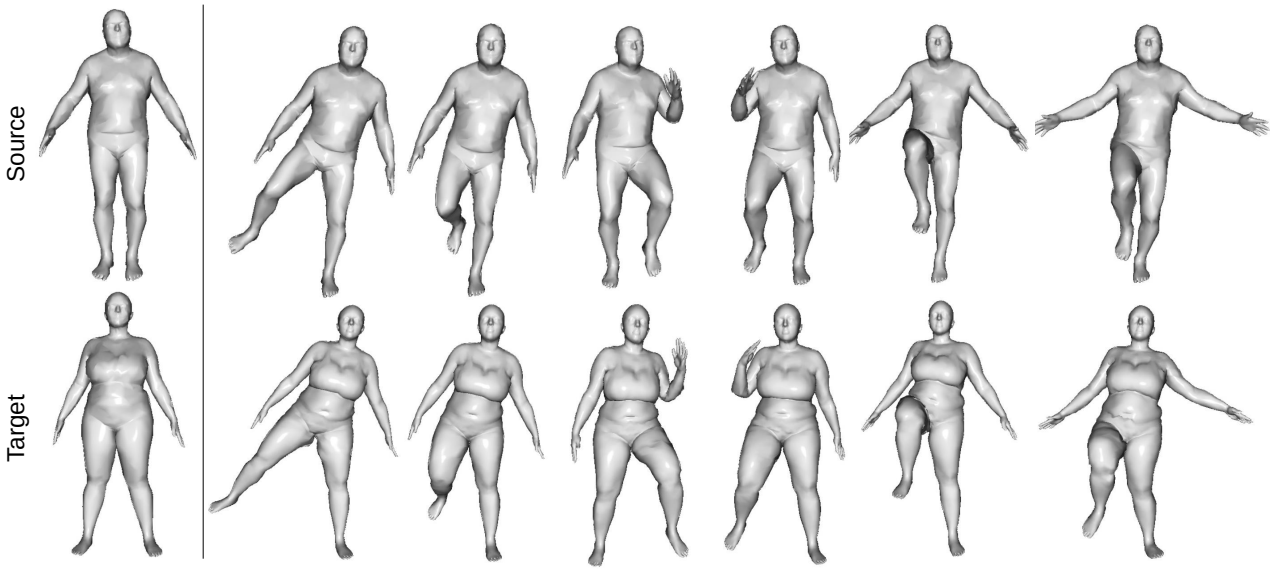

Figure 7. Human pose transfer results. The first column shows A_0 and B_0.

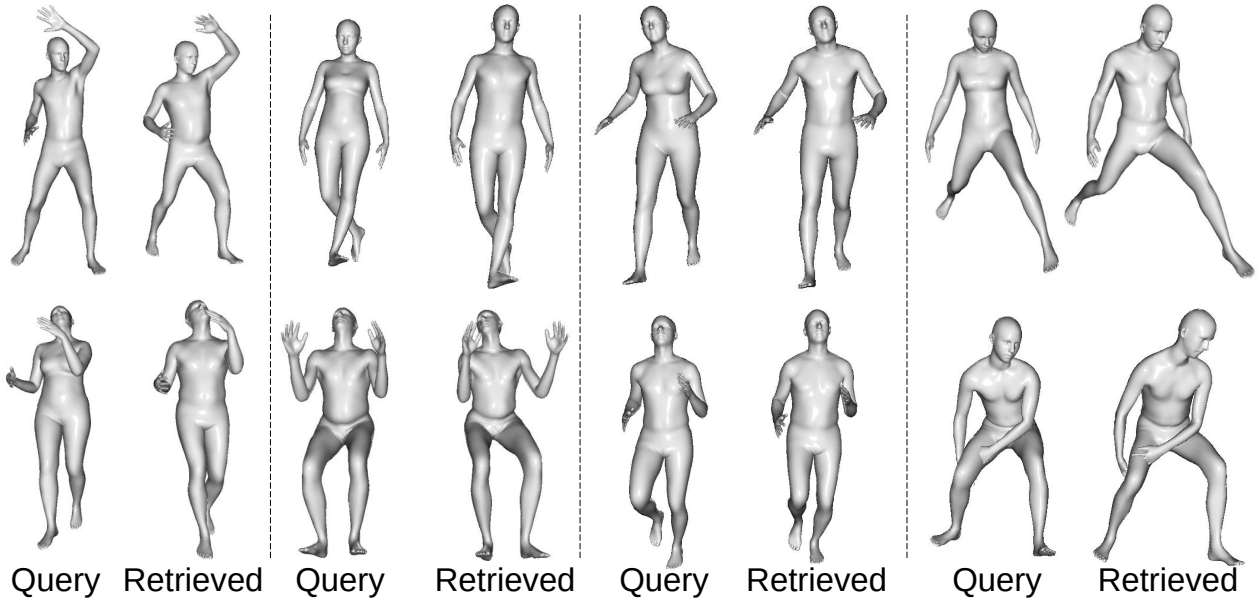

Figure 8. Results of shape retrieval on SURREAL 23k [17], where the query shapes are on the left and retrieved shapes are on the right.

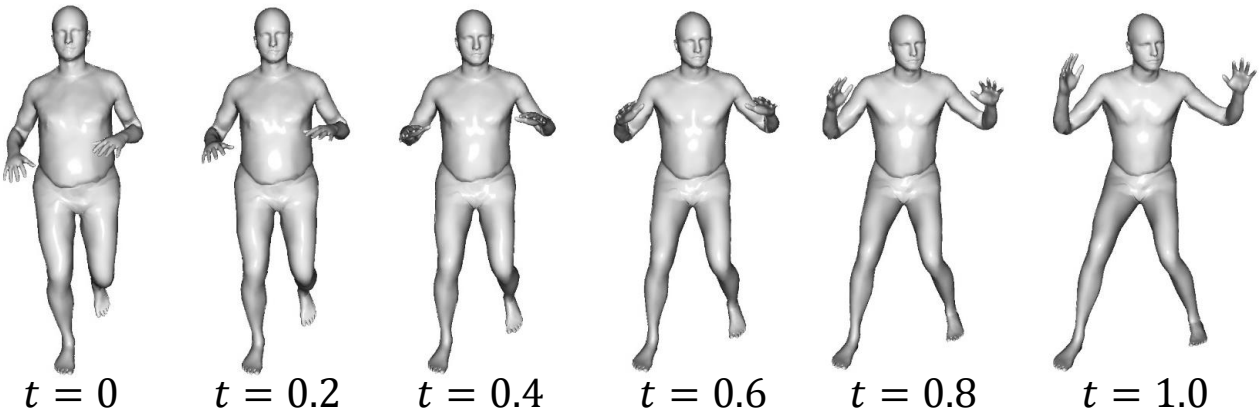

Figure 9. Shape interpolation results.

representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 3

[12] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[13] 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

[14] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM SIGGRAPH 2007 papers*, pages 80–es. 2007. 3

[15] Edgar Tretschk, Ayush Tewari, Michael Zollhöfer, Vladislav Golyanik, and Christian Theobalt. Demea: Deep mesh autoencoders for non-rigidly deforming objects. In *European Conference on Computer Vision*, pages 601–617. Springer, 2020. 1, 2

[16] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2

[17] 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. 3, 4, 5