Intrinsic Point Cloud Interpolation via Dual Latent Space Navigation: Supplementary material

Marie-Julie Rakotosaona[1] and Maks Ovsjanikov[1]

LIX, Ecole Polytechnique
{mrakotos,maks}@lix.polytechnique.fr

Abstract. In this document we collect some additional details about the proposed method, an ablation study and results, that due to lack of space were not included in the main manuscript.

1 Overview

In Section 2 we provide additional illustrations of our shape interpolation method. In Section 3 we demonstrate the performance of our approach for *shape reconstruction* highlighting the utility of our dual network for strong regularization of recovering high-quality shapes from noisy point clouds, as mentioned in the main manuscript. In Section 4 we provide an in-depth ablation study of our network design. In Section 5 we demonstrate the performance of our approach in the unsupervised case (when the training data is not in correspondence). In section 6, we develop intuitive connections to Riemannian geometry. Finally, in Section 7 we provide details of our architecture. Please note that we will release our full implementation upon potential acceptance. Note also that we provide a video as part of the supplementary materials.

2 Shape interpolation

2.1 Video and Comparison to Optimization-based Approaches

We provide a video which contains qualitative comparisons of interpolations on DFAUST and SMAL test sets with our main baselines. Note that our approach produces visually smoother interpolations with significantly lower distortions than all baselines across all shape pairs.

In the video we also provide comparisons with optimization-based approaches that achieve low distortion in Table 1 of the main manuscript. Specifically note that methods such as GD Coord. 1) require the input shapes to be in 1-1 correspondence 2) rely on expensive optimization at test time (for this reason, we compute these interpolations at half of the frame-rate), and most importantly 3), as shown the accompanying video, as they are not learning-based, lead to non-realistic intermediate shapes.

2.2 Additional Illustrations & Evaluation

In Figure 1 we provide an additional qualitative comparison of the linear interpolations in the basic shape (PointNet) AE latent space and the interpolation using our method. Our method preserves body type better (row 2) and interpolates well between a pair of shapes where the end result differs highly from the linear interpolation of the coordinates (row 4).

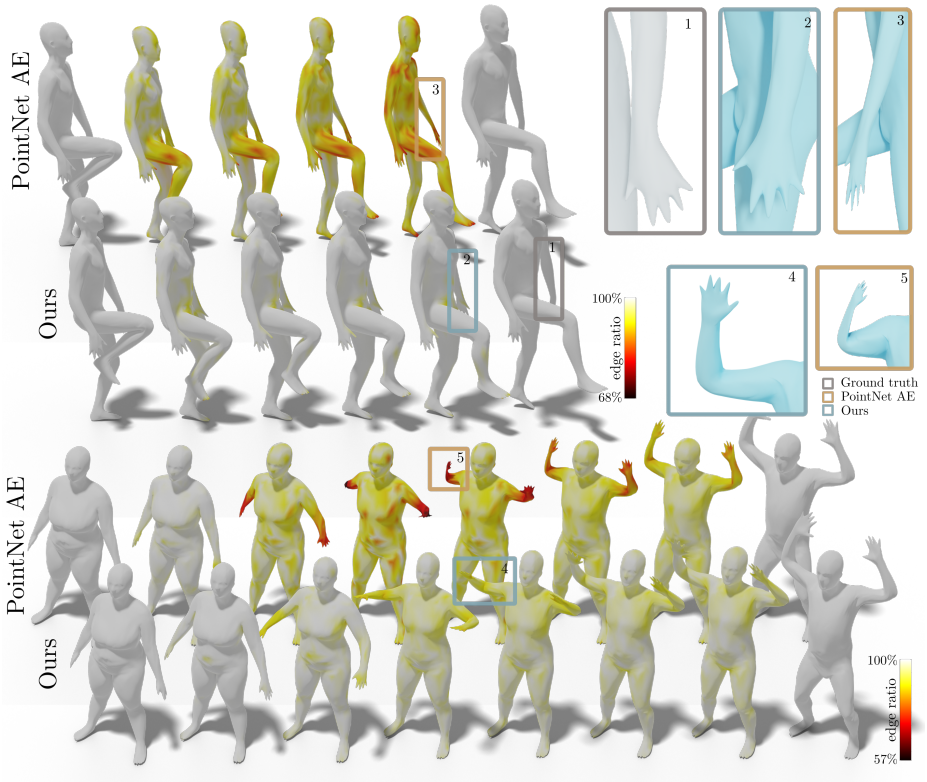

Fig. 1. We compare linear interpolations in PointNet AE latent space and interpolation using our approach. We visualize the ratio between the linear interpolation of edge lengths and edge lengths of the computed interpolations, to help highlight problematic areas.

We further compare our method to other baselines on the SMAL animals dataset. Table 1 reports the mean-squared variance of several shape features during interpolation of 100 pairs among 50 shapes obtained by farthest points sampling on this dataset. Note that our method produces significantly better quantitative results across all shape features.

	edge length	area (10^{-3})	volume (10^{-2})
PointNet	2.068	3.742	2.754
GD L2	1.906	3.618	2.681
GD EL	1.899	3.585	2.575
3D Coded	9.359	16.922	19.969
Ours	**1.538**	**2.975**	**1.728**

Table 1. MS variance of various shape features obtained from interpolating 100 pairs among 50 shapes obtained by farthest points sampling on animals dataset (SMAL)

We also test our method on *real scans* from the DFAUST dataset [3] in Figure 2. We observe that our method leads to more realistic results with lower distortion.

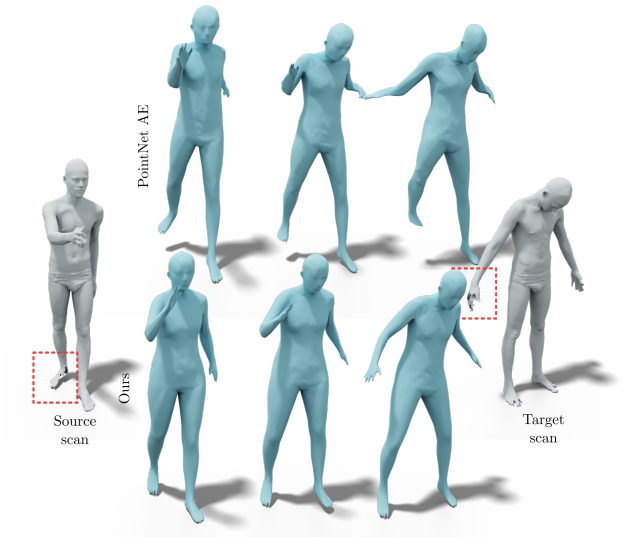

Fig. 2. We compare linear interpolations in PointNet AE latent space and interpolation using our approach on real scans with artefacts.

3 Shape reconstruction

As mentioned in the main manuscript, our approach not only enables better interpolation, but also results in more accurate reconstructions from noisy input. Here we provide additional qualitative and quantitative evaluation of the reconstruction performance and comparison to different baseline methods.

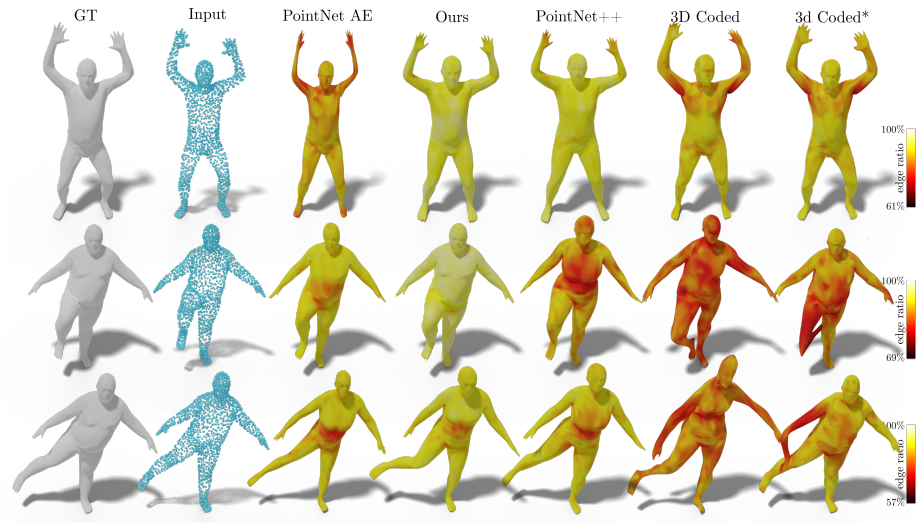

Fig. 3. Reconstruction of meshes from point clouds containing 1000 points, sampled from the underlying shape.

Recall that for our method, given a noisy unordered point cloud P, we reconstruct the shapes by using the following combination of our trained networks $\text{dec}_p(M_{EP}(M_{PE}(\text{enc}_p(P))))$, which differs from the standard auto-encoder approach $\text{dec}_p(\text{enc}_p(P))$. Therefore, in this section we show that the additional regularization provided by our mapping networks M_{EP}, M_{PE} results in better shape reconstruction.

To be fair to 3D-CODED, we normalize the total area of the output shapes. We evaluate this method before (3D-CODED) and after (3D-CODED*) their additional step of Chamfer Distance minimization. Note that in the case of 3D-CODED* additional optimization *at test time* is required to recompute the latent code that best approximates the input. Our method, on the other hand, performs the reconstruction in one shot.

In all of the experiments the training data is the combination of DFAUST and SURREAL datasets, and the test data is the DFAUST test shapes, both with and without noise.

Table 2 shows reconstruction results for several baselines on the 800 DFAUST test shapes. We report the edge length accuracy (EL), rotation-invariant point cloud reconstruction accuracy (PC) and per triangle area reconstruction accuracy (area). Note that our approach achieves the best overall reconstruction accuracy, especially on the intrinsic quantities and gives slightly worse reconstruction extrinsic loss (PC) compared to PointNet AE. We provide qualitative examples in Figure 3. Note that our method leads to both preservation of the overall shape structure and significantly less intrinsic distortion compared to all baselines.

	EL (10^{-5})	PC (10^{-4})	area (10^{-8})
PointNet AE	3.023	**2.120**	2.454
Edge Length AE	3.127	-	-
Ours $L_{1,2,3}$	**1.641**	2.572	**1.562**
3D-CODED	6.323	5.803	5.485
3D-CODED*	6.284	4.260	5.409
PointNet++	2.835	3.224	2.835

Table 2. Mean squared reconstruction losses on DFAUST testset. Edge length reconstruction loss (EL), Point cloud coordinates reconstruction loss (PC) and per triangle area difference

Table 3 (left) shows reconstruction performance on noisy point clouds. Note that we test using our model which was trained on clean data. Each noisy point cloud is obtained by adding Gaussian noise magnitude 5% of the scale of the mesh to each vertex coordinate. We observe that our method outperforms the other baselines for all the features. Figure 4 shows reconstructed meshes from the noisy point clouds. Notice that our method performs better at recovering the original pose and body type than the different baselines.

	Noisy dataset			Undersampled dataset		
	EL (10^{-5})	PC (10^{-4})	area (10^{-8})	EL (10^{-5})	PC (10^{-4})	area (10^{-8})
PointNet AE	5.663	8.538	5.650	3.847	**3.313**	2.810
Ours	**3.016**	**7.329**	**2.812**	**1.854**	3.587	**1.685**
3D-CODED	8.553	10.463	7.058	6.219	6.898	5.341
PointNet++	26.837	81.379	18.23	36.223	117.824	27.541

Table 3. Mean squared reconstruction losses on the DFAUST testset with noise (left) or undersampled (right). We use 5% of the shape bounding box gaussian noise on the testset. We randomly sample 500 points from the test shapes surfaces. We recall that the network was trained on 1000 point clouds. We show the edge length reconstruction loss (EL), the rotation invariant reconstruction loss (PC) and the per triangle area difference

Table 3 (right) shows reconstruction results on simplified point clouds. We randomly sample 500 points from the test shapes surfaces. We recall that the network was trained on 1000 point clouds. We observe that our method is more robust to under-sampling. In particular, and contrary to other methods, the intrinsic properties remain competitive with the performance from Table 2.

We also demonstrate the generalization power across different datasets by showing in Figure 6 examples of reconstructions from SCAPE dataset [1]. While the simple PointNet AE, is still able to reconstruct the overall position of the tested human, the output has distortions near the hands (left) and the legs (right). Our method generates more natural meshes even though the dataset is completely unknown with an entirely different underlying mesh, different body

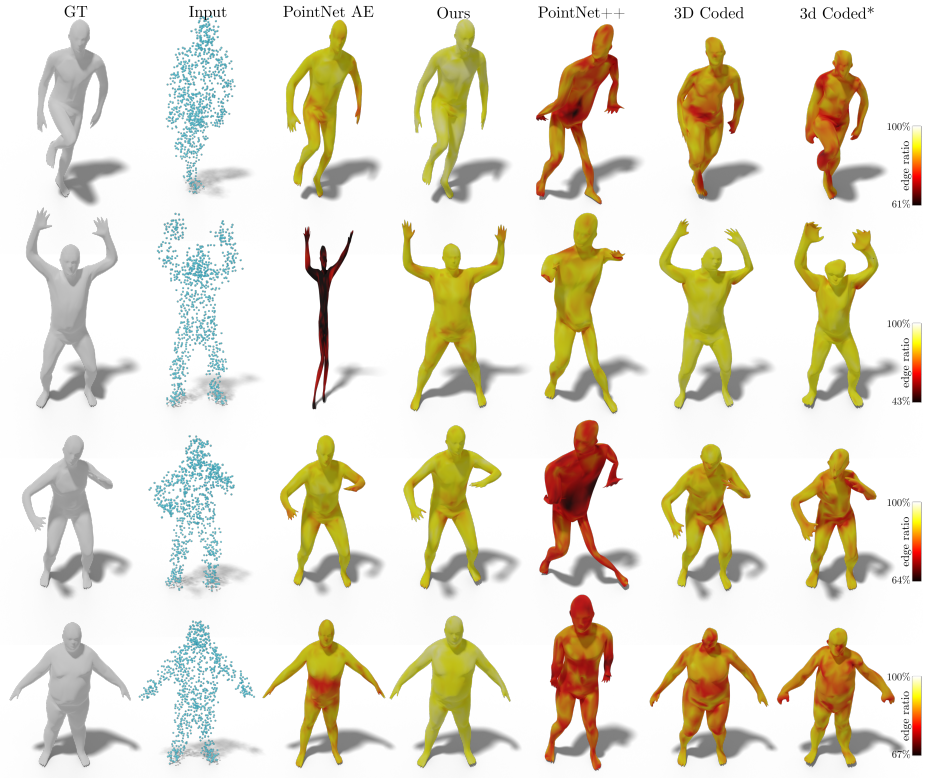

Fig. 4. Reconstructions from point clouds with 5% of the shape scale gaussian noise.

type and poses that are different to those seen at training. Note that we do not display the color coding as we do not have access to ground truth edge lengths.

4 Ablation study

4.1 Architecture design

Importance of multiple separate networks We first test the utility of having separate networks, rather than training a single network with a combined loss. Specifically, in our study, we have observed that introducing intrinsic information directly during the training of the shape auto-encoder produces unrealistic results with significant artefacts. (Fig. 7) We train two point-cloud AE (auto-encoders) using: a combination of edge (L_e) and point coordinate (L_{rec}) losses and edge (L_e), point coordinate (L_{rec}) and linearity losses (L_{lin})

Effect of separate networks training In our experiments, we fix the weights of the shape AE and edge auto-encoder during the training of the mapping net-

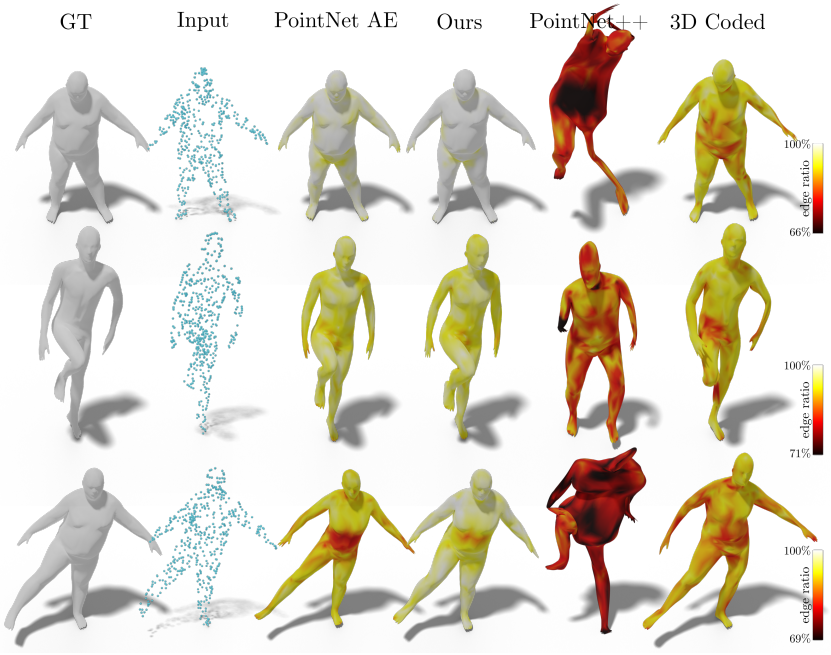

Fig. 5. We reconstruct a mesh from 500 points sub-sampled randomly from the ground truth mesh. We use a network pre-trained on inputs of size 1000 points.

works. By doing so, we fix the latent space and generating capabilities of each network. We believe that if this constraint is not respected, the shape AE and edge auto-encoder can be indirectly trained for different losses and generate distortions in the generated shapes. Here, we train the mapping networks, edge auto-encoder and shape AE at the same time. To make the training easier, we use a pretrained shape AE and edge auto-encoder. As seen in Table 4, the reconstruction losses are better than before. However, the shape AE can produce non natural reconstructions during interpolations as shown in Figure 8. We believe that if the shape AE and edge auto-encoder network were not pretrained, the resulting reconstructed shapes would present even more distortions since the pretrained shape AE can already generate decent natural looking shapes on parts of the dataset.

Auto-encoder vs Variational auto-encoder During our study we compared the performances of our pipeline using either a PointNet AE or a PointNet VAE. The type of network did not result in significant differences. By instance the mean squared variance of the edge length for our architecture trained with a VAE is 0.2301 and 0.2311 when trained with a AE (respectively 0.3760 and 0.3510 for the simple VAE and AE without using our pipeline).

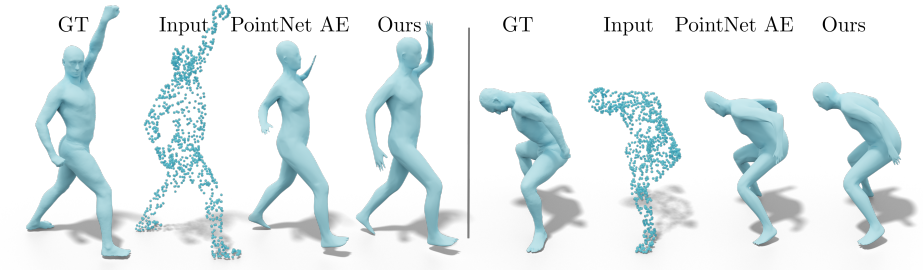

Fig. 6. Shape reconstruction from SCAPE. We reconstruct from 1k random points on the surface.

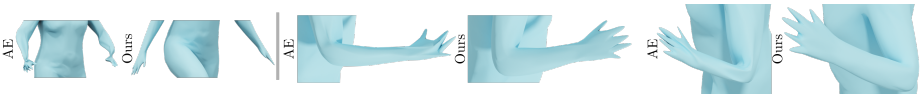

Fig. 7. Simple AE trained with L_e and L_{rec} (left) or L_e, L_{rec} and L_{lin} (right) produces artifacts during interpolation.

4.2 Choice of losses

Importance of cycle consistency loss. We train the mapping networks with direct reconstruction losses instead of cycle consistency losses as described in section 4.2 with L_{map1}, L_{map2}, L_{map3} :

$$\begin{aligned} L_{direct}(P, E_P) = \alpha d^{rot}(\text{dec}_p(M_{EP}(\text{enc}_e(E_P))), P) \\ + \beta \|el(\text{dec}_p(M_{EP}(\text{enc}_e(E_P)))) - E_P\|^2 \\ + \|\text{dec}_e(M_{PE}(\text{enc}_p(P))) - E_P\|^2 \end{aligned} \tag{1}$$

In Table 5, we observe that the quality of the map and the quality of the reconstructions are worse. In Figure 9 we show the cumulative distribution function of the edge length reconstruction loss on the testset. While most shapes seem to have reasonable edge reconstruction quality, outlier points make the reconstruction loss explode. Since cycle consistency is not enforced, the network

	EL (10^{-5})	PC (10^{-4})	area (10^{-8})
Ours	1.666	2.611	1.554
Ours sim. train.	1.027	1.464	1.027

Table 4. Mean squared reconstruction losses on the DFAUST testset. We present our main network and an alternative model where all three components are trained simultaneously. Edge length reconstruction loss (EL), Point cloud rotation invariant reconstruction loss (PC) and per triangle area difference (area).

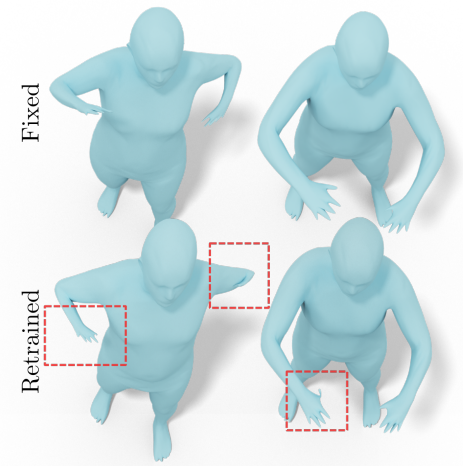

Fig. 8. Shape distortions are appearing during interpolation if the shape AE, edge auto-encoder and mapping networks are trained at the same time.

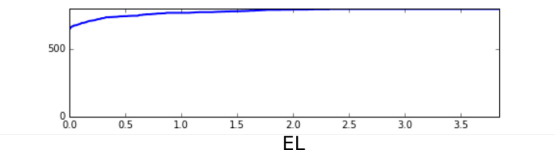

Fig. 9. Cumulative distribution function of edge reconstruction loss on the DFAUST testset for our network trained without cycle consistency with L_{direct}.

can map shapes onto outliers in the shape space that do not correspond to reasonable natural shapes.

Mapping losses In Table 6 we show an ablation study of the different losses combinations (described in section 4.2 of the main manuscript) used for training the mapping networks. The subscripts $1, 2, 3$ denote the use of L_{map1}, L_{map2}, L_{map3} respectively. We observe that when trained with L_{map2}, L_{map3}, so only intrinsic features, the model produces better intrinsic reconstruction performances to the expense of the extrinsic reconstruction loss. On the contrary, when trained with only L_{map1} and L_{map3} the network produces good point coordinate reconstruction but worse intrinsic reconstruction performances. To combine the benefits of the different losses, we choose to experiment with a model trained with the 3 losses.

	EL	PC	area
PointNet AE	$3.023 * 10^{-5}$	$2.120 * 10^{-4}$	$2.454 * 10^{-8}$
Ours	$1.641 * 10^{-5}$	$2.572 * 10^{-4}$	$1.562 * 10^{-8}$
Ours L_{direct}	0.1019	0.6289	$1.338 * 10^{-2}$

Table 5. Mean squared reconstruction losses on the DFAUST testset.

Cycle consistency and direct loss regularization Finally, we combine our cycle consistency loss with direct versions of L_{map1}, L_{map2}, L_{map3} described in equations from 1. In table 7, we observe that the models trained with cycle consistency only and cycle consistency with direct losses produce comparable results.

	EL (10^{-5})	PC (10^{-4})	area (10^{-8})
Ours $L_{2,3}$	1.595	14.816	1.490
Ours $L_{1,3}$	2.301	2.245	2.113
Ours $L_{1,2,3}$	1.641	2.572	1.562

Table 6. Ablation study on different mapping network losses. The subscripts $1, 2, 3$ refer to L_{map1}, L_{map2}, L_{map3} respectively. We show the mean squared reconstruction losses on DFAUST testset. Edge length reconstruction loss (EL), Point cloud coordinates reconstruction loss (PC) and per triangle area difference

	EL	area (10^{-4})	Volume (10^{-4})
Ours $L_{1,2,3}$	0.231	1.261	0.342
Ours $L_{1,2,3}$ with L_{direct}	0.342	1.315	0.264

Table 7. We report the mean squared variance of the edge length (EL), per surface area and total shape volume over the interpolations of 100 shape pairs. We compare our method, and our method trained with extra direct losses.

Linearity regularization term in edge auto-encoder. We train a version of our network without the linearity regularization term L_{lin} described in Eq. (6) of the main manuscript for training the edge auto-encoder. As seen in Table 8, the interpolations in the latent space of the edge auto-encoder are smoother when the network is trained with the linearity term. In Table 9, we observe that this term is also related to smoother interpolations of shapes.

	EL
Edge AE	0.199
Edge AE no lin. reg.	1.777

Table 8. We report the mean squared variance of the edge length (EL) over the interpolation in the edge length AE latent space of 100 shape pairs.

	EL	area (10^{-4})	volume (10^{-4})
Ours	0.230	1.220	0.385
Ours no lin. reg.	0.245	1.361	0.430

Table 9. Interpolation losses for our network where the edge auto-encoder is trained with and without linearity regularization term. We report the mean squared variance of the edge length (EL), per surface area and total shape volume over the interpolations of 100 shape pairs from the DFAUST testset.

5 Interpolation in the unsupervised case

Our method can be adapted to an unsupervised context where the 1-1 correspondences are not provided during training. The training process can be described in 3 steps: We first train a point cloud auto-encoder that takes unordered point clouds and outputs an ordered point clouds where the order corresponds to given template T. Then we train the edge auto-encoder by using the output of the shape auto-encoder as training data. Finally, we train the mapping networks as described in the main manuscript.

We first initialize the weights by pre-training the shape AE network to output a chosen template mesh using a variant of the reconstruction loss L_{rec} described in Eq. 4 of the main manuscript.

$$L_{recInit}(P) = \frac{1}{n}\sum_{i=1}^{n} \|T_i - \tilde{P}_i\|^2, \text{ where } \tilde{P} = \mathrm{dec}_p\left(\mathrm{enc}_p(P)\right). \tag{2}$$

Then we train the model using Chamfer Distance (CD) from Eq. (3) while encouraging the network to maintain the learned triangulation from step 1 by using regularization terms similar to those used in [4] described bellow.

$$CD(\tilde{P}, P) = \frac{1}{n}\sum_{p_i \in \tilde{P}} \min_{p_j \in P} \|p_i - p_j\|_2^2 + \frac{1}{n}\sum_{p_j \in P} \min_{p_i \in \tilde{P}} \|p_j - p_i\|_2^2 \tag{3}$$

$$L_e^{reg}(E_{\tilde{P}}) = \|E_{\tilde{P}} - E_T\|_2^2, \text{where } \tilde{P} = \mathrm{dec}_p(\mathrm{enc}_p(P)) \tag{4}$$

$$L_{lap}^{reg}(\tilde{P}) = \|L * (\tilde{P} - T)\|_2^2, \text{where } L \text{ is the graph laplacian} \tag{5}$$

We compare our method to unsupervised versions of PointNet AE and 3D Coded. We report numerical evaluation of the interpolations in Table 10. Note, that our method leads to improved shape features compared to other methods. In Figure 10, we observe that our method produces more realistic shapes, in particular it produces better arms and heads than PointNet AE and better arms than 3D Coded.

	EL	area (10^{-4})	volume (10^{-5})
3D Coded (unsupervised)	0.982	4.140	16.054
PointNet AE (unsupervised)	0.597	3.508	5.251
Ours (unsupervised)	**0.398**	**2.752**	**4.718**

Table 10. We report the mean squared variance of the edge length (EL), per surface area and total shape volume over the interpolations of 100 shape pairs. We highlight, while all models produce worse results than their supervised equivalents, our method leads to better interpolations.

6 Geodesics in non flat domains

As mentioned in the main manuscript the two auto-encoders of our architecture can be interpreted as parametrizing the space of realistic shapes and endowing this space with metric (distance) structure. Specifically, the shape auto-encoder aims to recover realistic 3D shapes, and we can *compare* shapes by computing the Euclidean distance between their associated latent vectors in the edge-length auto encoder. Below we explore the relation between linear interpolation and geodesic paths on curved surfaces.

First we note that a classical result in differential geometry (a consequence of Gauss's Theorema Egregium) states that it is impossible to parametrize a curved surface using a Euclidean coordinate system while mapping geodesic paths to straight line segments [2] (Chapter 3.1). This directly implies (up to mild genericity conditions such as smoothness) that *there does not exist an auto-encoder network* that is both bijective onto some latent space and allows to recover geodesics through linear interpolation of the latent vectors. Said differently, linear interpolation in the latent space only allows to recover a *flat metric* on the space of shapes, while the intrinsic distortion metric can induce curvature in shape space [5].

Nevertheless, we observe that in certain cases linear interpolation *can* be used to recover geodesic paths even for non-flat domains, if the shape is embedded into a larger space. Specifically, consider the standard sphere S^{n-1} embedded in $\mathbb{R}^n$ and two points $p, q \in S^{n-1}$ that are not polar opposites. Now, construct a line segment linearly interpolating p, q in $\mathbb{R}^n$ and then *project* this line segment onto S^{n-1}. It is clear that the projected segment will recover the geodesic path on the sphere, despite using a linear interpolation in Euclidean space.

This simple example illustrates that if a surface is embedded in a larger space (so that the map from the surface to this space is not a parametrization as it is not invertible) points in that space can be mapped onto the surface through projection. While the projection will necessarily introduce distortion, it can nevertheless help recover geodesic paths by providing projected points informed by the metric in the embedding space. Although simple and very special, this example points at the interest in studying the relation between the latent spaces of auto-encoders and Riemannian metrics, which we leave as exciting direction for future work.

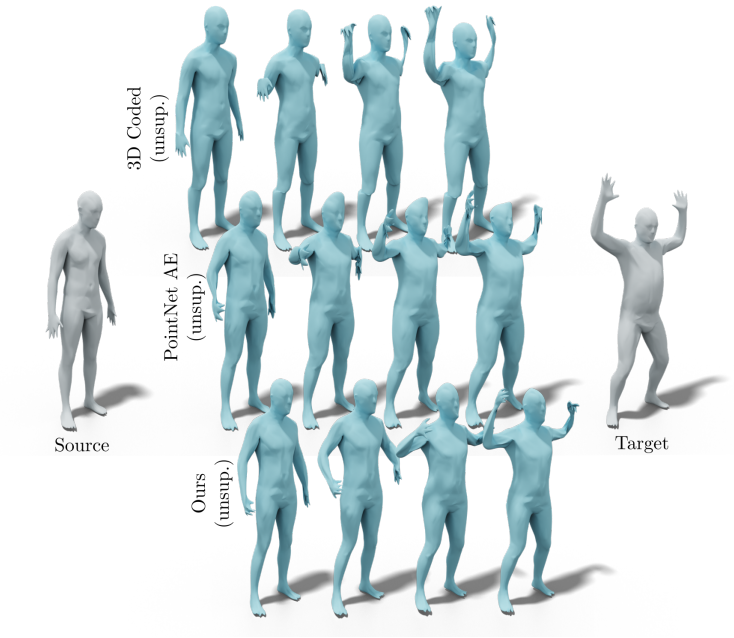

Fig. 10. Interpolation between shapes when trained with no 1-1 correspondences at train time. Our method produces more realistic shapes.

7 Architecture details

We present the detailed architecture of the shape AE, edge length AE and mapping networks in Figure 11, 12, 13.

We implemented the presented architectures using Tensorflow and the Adam optimizer for training. Our complete implementation will be released upon acceptance.

References

1. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM transactions on graphics (TOG). vol. 24, pp. 408–416. ACM (2005)
2. Berger, M.: A panoramic view of Riemannian geometry. Springer Science & Business Media (2012)
3. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: Registering human bodies in motion. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)
4. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3d-coded: 3d correspondences by deep deformation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 230–246 (2018)

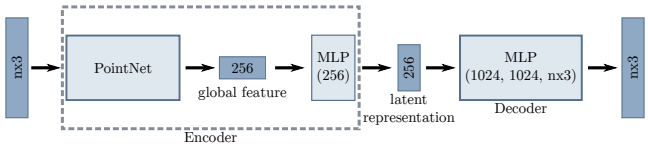

Fig. 11. Shape AE architecture.

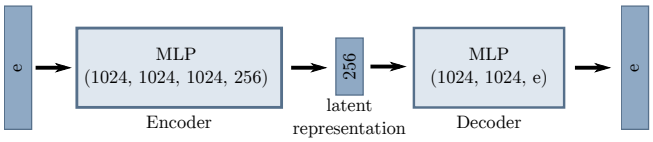

Fig. 12. Edge length AE architecture.

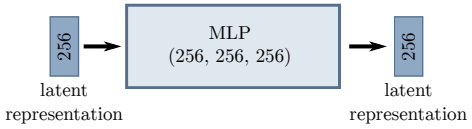

Fig. 13. Mapping networks architecture.

5. Heeren, B., Rumpf, M., Schröder, P., Wardetzky, M., Wirth, B.: Exploring the geometry of the space of shells. In: Computer Graphics Forum. vol. 33, pp. 247–256. Wiley Online Library (2014)