OperatorNet: Recovering Shapes from Difference Operators -Supplementary Material-

Ruqi Huang*
LIX, Ecole Polytechnique
rqhuang88@gmail.com

Marie-Julie Rakotosaona*
LIX, Ecole Polytechnique
mrakotos@lix.polytechnique.fr

Panos Achlioptas
Stanford University
optas@cs.stanford.edu

Leonidas Guibas
Stanford University
guibas@cs.stanford.edu

Maks Ovsjanikov
LIX, Ecole Polytechnique
maks@lix.polytechnique.fr

Overview We present supplementary results and verification of our OperatorNet. In Section A, we demonstrate some examples of shape analogies that conduct gender transfer (Figure 1), pose and style transfer across human shapes (Figure 2), and animal analogies (Figure 3). Then in Section B, we add more baseline comparisons for shape interpolation including PointNet++ [1], MultiChart [2] and 3D-Coded [3] (Figure 4), and we show that interpolating shape differences using multiplicative scheme is better than the naive linear scheme (Figure 5). In Section C, we verify the generalization power of OperatorNet (Figure 6), and also show more complete results of reconstructing shapes with different triangulations (including a geometric baseline comparison). The ablation study on the network depth and shape difference input combinations is presented in Section D. Finally, in Section E we present the proof of Theorem 1 in our main paper.

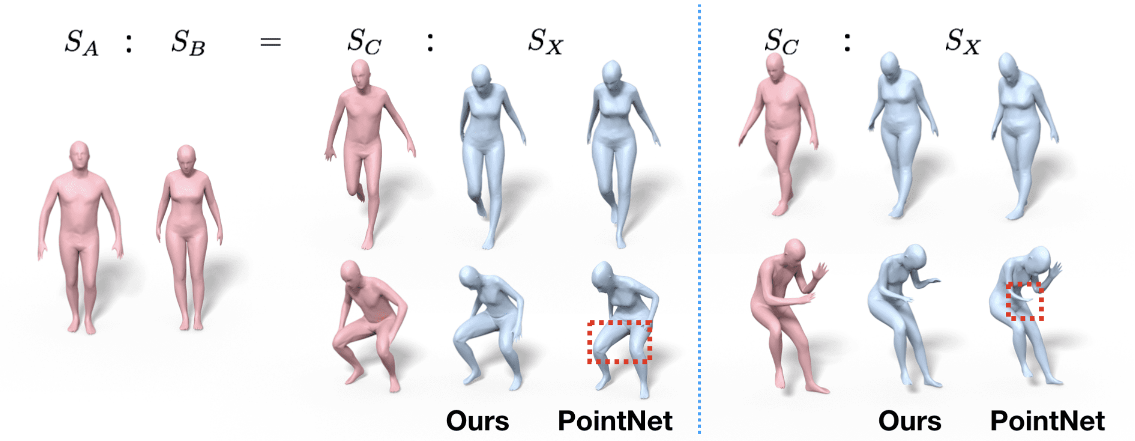

Figure 1: Gender analogies via OperatorNet and PointNet. Note that though in some cases PointNet also delivers reasonable results (e.g. the ones on the top row), the results of OperatorNet are more natural and semantically meaningful (see, e.g., the discrepancies highlighted in the red dotted boxes).

A Shape Analogies

In addition to Figure 10 of our main paper, we present more gender analogies in Figure 1. Note that though in some cases PointNet also delivers reasonable results (e.g. the ones on the top row),

*denote equal contribution.

the results of OperatorNet are in general more natural and semantically meaningful (see, e.g., the discrepancies highlighted in the red dotted boxes).

We also present a set of shape analogies that transfer pose (top row) and style (bottom row) across human shapes in Figure 2. We observe that our results (the fourth column from the left) are both more natural and intuitive while PointNet (the right-most column) produces less satisfactory results with, e.g., local area distortions (see the red dotted boxes).

Lastly, we show analogies among animals in Figure 3, where we present both pose transfer (top row) and style transfer (bottom row) and comparison to the results of PointNet.

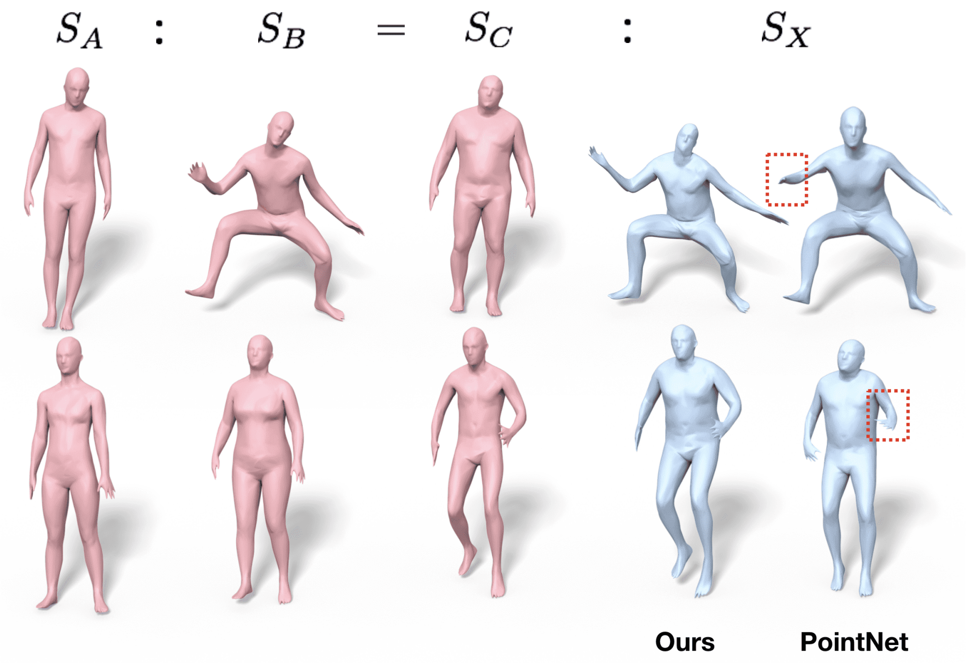

Figure 2: Human shape analogies via OperatorNet and PointNet auto-encoder (see the red dotted boxes for the discrepancies).

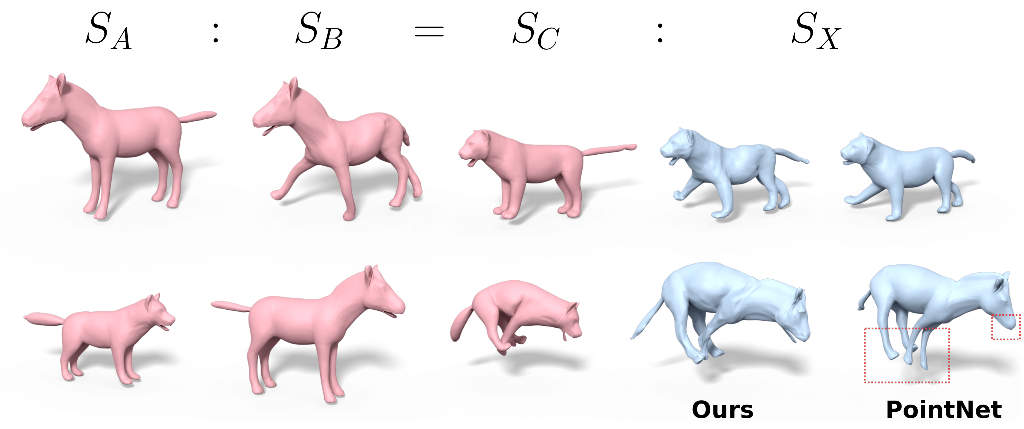

Figure 3: Top row: transferring the pose of S_B, from S_C to S_X. Bottom row: transferring the animal type of S_B, from S_C to S_X. PointNet does not maintain the correct pose (bottom row) and does not transfer details such as open mouths correctly.

B Shape Interpolation

Baseline Comparison To make our comparison more complete, we further compare our method to the auto-encoder proposed in 3D-Coded [3], Multi-Chart GAN proposed in [2], and a Point-

Net++ [1] based auto-encoder.

Regarding 3D-Coded [3] method, we first reconstruct the source and target shapes using their pre-trained model and linearly interpolate the produced latent representations. On the other hand, in [2], a GAN is trained to generate realistic human shapes. In particular, we follow the interpolation scheme described in [2]: first we pick two randomly generated latent vectors z_1, z_2, which, via the GAN give arise to two shapes $G(z_1), G(z_2)$. Then, the interpolation between the two shapes is achieved as $G(z(t))$, where $z(t) = (1-t)z_1 + tz_2$. We randomly generate 1000 shapes using their trained model and pick $G(z_i), i = 1, 2$ that are nearest to the end shapes in the bottom row of Figure 4. Lastly, similar to the PointNet baseline we report in the main paper, we train an auto-encoder with the PointNet++ encoder and our decoder.

The first row shows the interpolation of [3]. This method generates significant distortions during the interpolation, particularly on the arms. In the second row, note that the interpolations from Multi-Chart GAN [2] between the two end shapes are not evenly spaced. For instance, the arms change abruptly during the three middle shapes, while there is little change on that region afterwards. As seen in the fourth row of Figure 4, the result of PointNet++ based auto-encoder suffers similar distortions on the arms as that of PointNet (see the third row). The remaining rows have been shown and analyzed in the main paper (see Figure 9 therein).

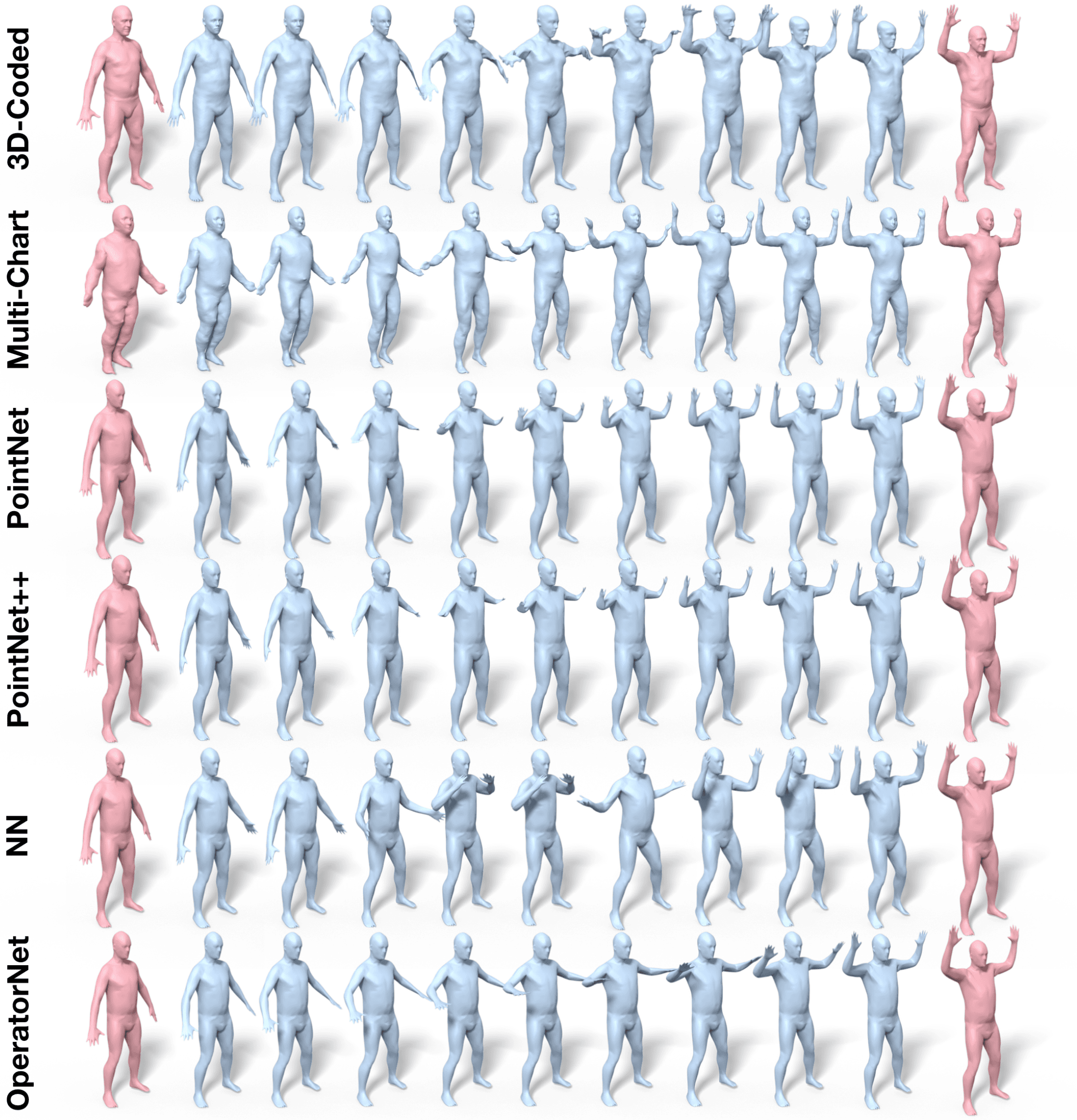

Figure 4: From the top row to the bottom row: interpolations via 3D-Coded, Multi-chart GAN, Pointnet, PointNet++ based auto-encoder, Nearest Neighbor in latent space and OperatorNet .

Linear Interpolation vs. Multiplicative Interpolation We note that, since the shape differences are represented by matrices, it is also possible to interpolate shape differences linearly, i.e., $\mathbf{D}(t) = (1-t)\mathbf{D}_0 + t\mathbf{D}_1$. However, as we argue in Section 3 in the main paper, the *multiplicative* property of shape differences suggests that it is more natural to interpolate the difference operators following Eq.(6) in our main paper. To illustrate this point, we show in Figure 5 interpolated sequences with respect to the two schemes above – the multiplicative one in the first row and the linear one in the second row. It is visually evident that the former leads to more continuous and evenly deformed sequence. Moreover, we compute the distance between consecutive shapes in both sequences and plot the distributions in the bottom panel of Figure 5 as a quantitative verification.

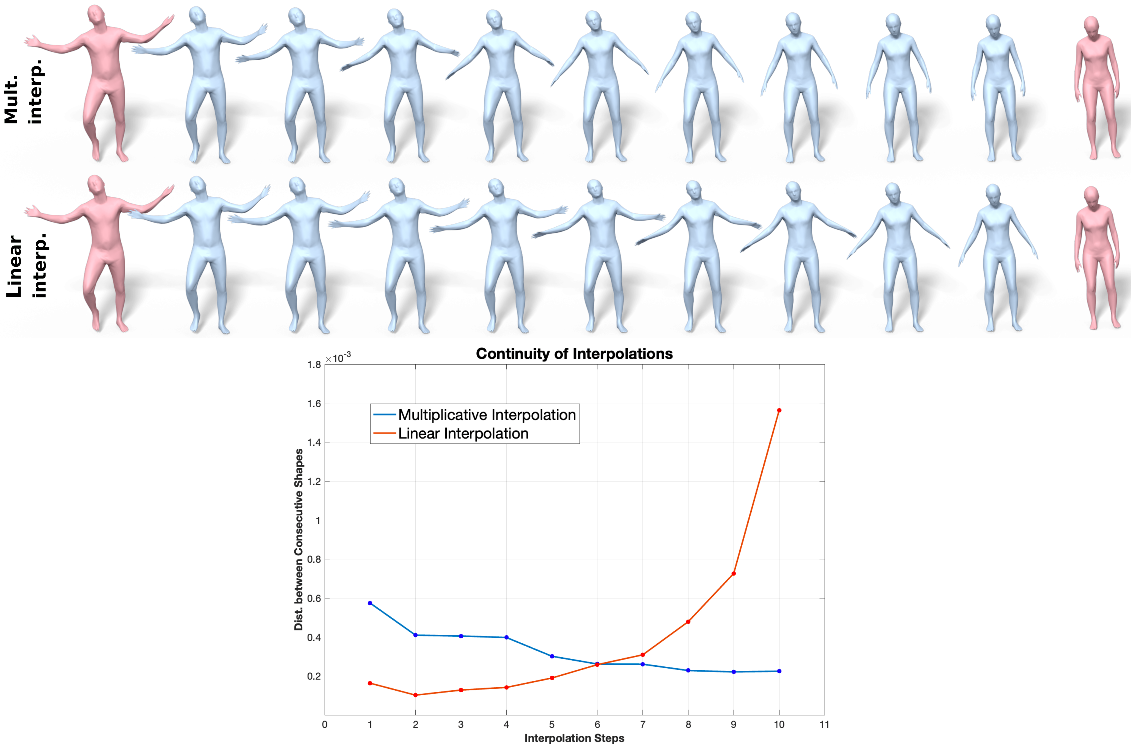

Figure 5: Reconstructions with respect to shape differences interpolated using multiplicative scheme (first row) and using linear scheme (second row). In the bottom panel we plot the distances between consecutive reconstructed embeddings for both sequences. The multiplicative scheme delivers a more smooth deformation sequence.

C Shape Reconstructions

Verification of Generalization Power of OperatorNet To demonstrate the generalization power of OperatorNet, we show in Figure 6 our reconstructions of test shapes from the SURREAL dataset. For comparison, we retrieve the shapes in the training set, whose shape differences are the nearest to the ones of the test shapes. In each of the figures, the top row presents the ground-truth test shapes; the middle row shows reconstructions from OperatorNet; the bottom row demonstrates the shapes retrieved from the training set via nearest neighbor search in the space of shape differences.

It is evident that OperatorNet accurately reconstructs the test shapes, which deviate from the shapes in the training set significantly, suggesting that our network generalizes well in unseen data.

Reconstruction of Shapes in Different Discretizations We show the reconstructions of shapes in a different discretization than the base shape in Figure 7, part of which (the top two rows) is demonstrated in Figure. 7 of the main paper. Here we further train an OperatorNet with finer labels (of 2k vertices compared to that of 1k vertices used in the original version) and show the reconstructions on the third row of Figure 7. We emphasize that the use of coarse labels is for a

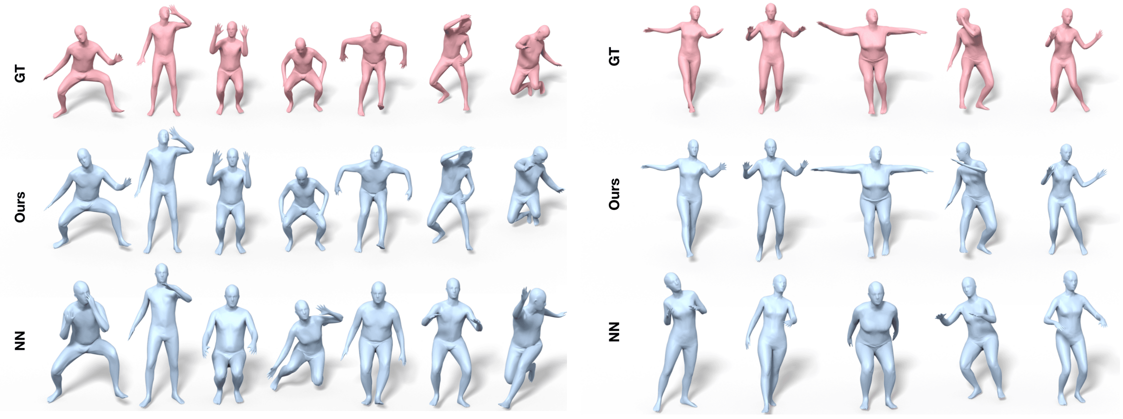

Figure 6: Top row: ground-truth embeddings; middle row: reconstructions via OperatorNet; bottom row: shapes from the training set, whose shape differences that are closest to the ones of the test shapes in the top row.

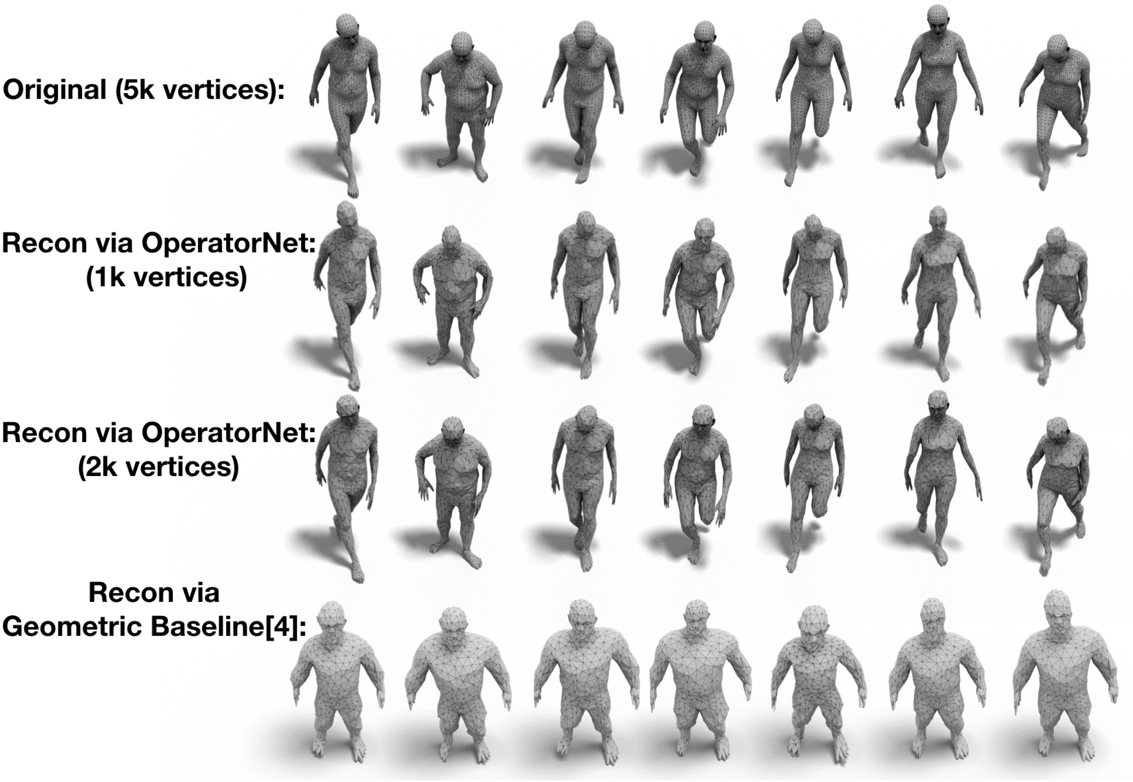
Original (5k vertices):
Recon via OperatorNet: (1k vertices)
Recon via OperatorNet: (2k vertices)
Recon via Geometric Baseline[4]:

Figure 7: Top row: input shapes with different number of vertices (5k) than that of the base shape (1k); second row: reconstructions of the original OperatorNet proposed in the main paper; third row: reconstructions of OperatorNet trained with higher resolution labels (2k vertices); bottom row: reconstructions via the geometric approach [4].

fair comparison to the geometrical baselines for reconstructing embeddings from shape differences. As shown in the third row, OperatorNet reconstructs the shapes in a higher resolution well, which is not possible for the geometric approaches.

Reconstructing shapes of different triangulations is extremely difficult for geometric approach: we demonstrate the reconstructions via the geometric approach [4] in the bottom row: the outputs are all close to the source shape (i.e., the base shape), which suggests that the algorithm struggles to find the right direction to deform the source to the target.

D Ablation Study on Network Design

We investigate multiple architectures for OperatorNet. In Table 1 we compare the reconstruction performance over different combinations of input shape differences, and different depths of encoders.

We report the performance of 4 different convolutional encoders from 1 to 4 layers deep by doubling the number of neurons every layer.

Two trends are observed in Table 1: first, we always achieve the best performance when all three types of shape differences are used, for varying depths of the network; second, fixing the combination of input shape differences, the network performs better as its depth gets shallower.

Putting these two observations together, we justify our final model, which has one single layer convolutional encoder and uses all three types of shape differences as input.

Table 1: Ablation study: auto-encoder performance on DFAUST testset (measured by the loss function as defined in Eq.(9) in the main paper, the errors in the table are at the scale of 10^{-4}).

Encoder architecture	Area	Ext	Conf	A+E	A+C	E+C	A+E+C
Conv. 8	**8.61**	**4.29**	**3.78**	**3.82**	**3.41**	**2.56**	**2.46**
Conv. 8×16	9.08	4.54	4.28	4.65	3.93	3.10	**3.05**
Conv. 8×16×32	9.90	5.54	4.91	5.59	4.88	3.71	**3.55**
Conv. 8×16×32×64	11.16	6.39	5.93	6.89	5.42	4.35	**4.24**

E Proof of Theorem 1.

Proof 1 *Since X is known to be of rank 3, and $\mathbf{G}$ is symmetric, we have, by SVD:*

$$\mathbf{G} = \mathbf{\Phi}^T A X X^T A \mathbf{\Phi} = U \Sigma U^T,$$

where, U, Σ are respectively the top 3 singular vectors and singular values of $\mathbf{G}$. Therefore, we have $\mathbf{\Phi}^T A X R = U\sqrt{\Sigma}$, where R is a 3×3 rigid transformation matrix satisfying $R^T R = I_{3\times 3}$. In other words, we recover $\mathbf{\Phi}^T A\tilde{X}$ from E^G, where $\tilde{X} = XR$ is equivalent to X up to rigid transformations. Then, to recover the projection of $\tilde{X}$ in the space spanned by $\mathbf{\Phi}$, we simply compute $\mathbf{\Phi}\mathbf{\Phi}^T A\tilde{X}$.

References

[1] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *CoRR*, vol. abs/1706.02413, 2017.

[2] H. Ben-Hamu, H. Maron, I. Kezurer, G. Avineri, and Y. Lipman, "Multi-chart generative surface modeling," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 215:1–215:15, Dec. 2018.

[3] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry, "3d-coded : 3d correspondences by deep deformation," in *ECCV*, 2018.

[4] E. Corman, J. Solomon, M. Ben-Chen, L. Guibas, and M. Ovsjanikov, "Functional characterization of intrinsic and extrinsic geometry," *ACM Trans. Graph.*, vol. 36, no. 2, pp. 14:1–14:17, Mar. 2017.