Laplacian Mesh Transformer: Dual Attention and Topology Aware Network for 3D Mesh Classification and Segmentation – *Supplemental Material* –

Xiao-Juan Li[1,2], Jie Yang (✉)[1,2], and Fang-Lue Zhang[3]

[1] Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] Victoria University of Wellington
{lixiaojuan,yangjie01}@ict.ac.cn fanglue.zhang@vuw.ac.nz

OVERVIEW

This supplementary material accompanies the main paper, which presents data preparation, more implementation details, more quantitative evaluations on shape part segmentation and classification on other public datasets, more visualization results of shape part segmentation, and attention maps of dual attention.

All sections are organized as follows:

- Section 1 provides more details on data preparation and implementation.
- Section 2 provides more evaluations on shape classification on other datasets (*e.g.* SHREC11 [13], Cube Engraving [9]).
- Section 3 provides more evaluations on shape part segmentation on Human Body dataset [15].
- Section 4 provides more evaluations on the shape segmentation for the performance of the noisy input and with different number of vertex. And also for a fair comparison, we evaluate our method, PD-MeshNet [17], and MeshCNN [9] on different metrics on edges, faces, and vertices.
- Section 5 and Section 6 provide more visualization on shape part segmentation and importance maps of our proposed dual attention mechanism.

1 Implementation Details

In this section, we provide the statistics and more details of the datasets used in our experiments. Table 1 describes more details of the training and test for the application tasks, *i.e.* Classification (Cls.) or Segmentation (Seg.). We follow the available official training set and test set to split our data. For the SHREC11 dataset, there are 16 training shapes in each category, so we follow the default setting to perform the evaluation on the experiments, denoted as SHREC11-Split16. As for Split10, we randomly select 10 examples from the training set of each category. The reported performance is the average score

DataSet	ModelNet		ShapeNet	COSEG			Human Body	Cube Engraving	SHREC11	
	ModelNet10	ModelNet40		Vases	Chairs	Aliens			Split10	Split16
Task	CLS. (10)	CLS. (40)	SEG. & CLS. (16)	SEG.	SEG.	SEG.	SEG.	CLS. (22)	CLS. (30)	CLS. (30)
#objects	4889	12246	16003	300	397	200	399	4381	600	420
#training objects	3981	9791	11915	255	337	170	381	3722	480	300
#test objects	908	2455	4088	45	60	30	18	659	120	120

Table 1: **Dataset Statistics.** We summarize the data statistic of the datasets used in our experiments. For the Split10 of SHREC11, We report the average accuracy on three different random splits of training and test set. The number next to the task's name means the total number of categories in that dataset.

for the three random samples on the training set. For each shape, we adopt the manifold tool [11] to simplify the watertight meshes to around 2k vertices. Then, we obtain the first 12 Laplacian eigenvectors by Laplacian spectral decomposition [20]. Finally, our network takes the 12-d Laplacian eigenvectors and 3-d vertex positions as inputs for training our proposed networks. For the detailed algorithms of watertight manifold creation and Laplacian spectral decomposition, please refer to [11,20].

2 More Evaluation on Mesh Classification

We conduct more mesh classification experiments on two other datasets, SHREC11 [13] and Cube Engraving [9], to validate our method. SHREC11 contains 30 categories and 600 shapes, and Cube Engraving contains 22 categories and 4381 shapes. Specifically, we follow the previous work [10,9] to conduct two experiments, Split16 and Split10, on the SHREC11 dataset. In the Split16 experiment, we use the default setting to evaluate the performance of our model, where the training set of each category has 16 shapes. Similarly, each category contains 10 shapes in Split10. We then randomly select 10 shapes

Methods	SHREC11		Cube Engraving
	Split16	Split10	
GWCNN [5]	96.6	90	-
PointNet++ [19]	-	-	64.3
MeshCNN [9]	98.6	91.0	92.2
PD-MeshNet [17]	99.7	99.1	94.4
MeshWalker [12]	98.6	97.1	98.6
DiffusionNet [21]	-	99.7	-
SubdivNet [10]	**100**	**100**	**100**
Ours	**100**	**100**	99.7

Table 2: **Comparison on Mesh Classification for SHREC11 [13] and Cube Engraving [9] datasets.** We report the accuracy of classification. Note that '-' indicates the number is not reported in their paper.

for each category and report the average score for the top-3 selections. Table 2 presents the comparison with other alternative methods on the two datasets. Our method and [10] achieve the best performance (accuracy **100%**) on SHREC11. On the Cube Engraving dataset, our performance of classification is comparable with the SOTA [10].

3 More Evaluations on Human Body Segmentation

We perform one more evaluation on human body segmentation on the dataset proposed in [15]. The dataset is composed of 381 training shapes from SCAPE [2], FAUST [3], MIT [22], Adobe Fuse [1], and 18 test shapes from SHREC07 [7]. All the shapes are segmented into 8 parts. Since the semantic labels are defined on the edges of the mesh, we transfer the edge-wise labels to vertex-wise labels. Followed by [10], all the meshes are scaled into a unit sphere and re-meshed by Liu *et al.*'s method [14] to ensure lower distortion of details. Results are reported in Table 3, showing that our approach outperforms the others. For the method in [10], we take the reported results directly from their paper and list them in Table 3. According to their results, our proposed method presents a superb performance over the other alternative methods. Figure 4 shows some segmented results on the human body shapes.

4 More Evaluations on Part Segmentation

In this section, we evaluate the robustness of our method and compare our method and other alternatives in a fair manner.

For the evaluation of robustness, we use our pretrained model with noisy input and different numbers of vertices. The performance is reported in Table 4. For the noisy input, we choose two way to add some noises into the input data, adding Gaussian noises on the surface and generating holes by removing some faces. For adding Gaussian noises on the surface, we added different translations (0.05,0.1,0.2) along the normal of each vertex to generate noises on the test data. For removing faces, we randomly removed the triangles with different percentages (10%, 20%, 50%) on the test data. We then fed them to our pre-trained

Method	Accuracy
PointNet [18]	74.7
PointNet++ [19]	82.3
Toric Cover [15]	88.0
MeshCNN [9]	87.7
PD-MeshNet [17]	86.9
SNGC [8]	91.3
MeshWalker [12]	92.7
SubdivNet [10]	93.0
DynGraphCNN [23]	89.7
GCNN [16]	86.4
DiffusionNet [21]	91.7
Ours	**93.9**

Table 3: **Mesh segmentation accuracy on the human body dataset [15]**. From the results, we can observe that our network achieves the best performance on the dataset compared to alternative approaches.

Method	Vases	Chairs	Tele-alines	Mean
Ours-noise(0.05)	98.0	97.7	97.3	97.6
Ours-noise(0.10)	97.7	97.5	97.0	97.4
Ours-noise(0.20)	96.8	97.0	96.1	96.6
Ours-hole1(10%)	97.9	97.4	97.1	97.4
Ours-hole2(20%)	93.2	93.7	92.7	93.2
Ours-hole3(50%)	87.6	88.1	85.9	87.2
Ours-test(4k)	98.0	97.5	**97.7**	97.7
Ours-test(1k)	97.8	97.1	96.9	97.1
Ours(2k)	**98.1**	**97.7**	**97.4**	**97.7**

Table 4: **Robustness Validation.** We evaluate the robustness of our method for the noisy input and different vertex numbers during inference.

	Method	Vases	Chairs	Tele-alines	Mean
Vertex	MeshCNN	85.2	92.8	94.4	90.8
	PD-MeshNet	81.6	90.0	89.0	86.9
	Ours	**98.1**	**97.7**	**97.4**	**97.7**
Face	PD-MeshNet	95.4	97.2	**98.2**	96.9
	Ours	**98.7**	**98.0**	97.8	**98.1**
Edge	MeshCNN	97.3	**99.6**	97.6	98.1
	Ours	**97.9**	98.3	**98.2**	**98.1**

Table 5: **Comparison on different metrics.** For a fair comparison, we report the performance on the different metrics for three approaches, including PD-MeshNet [17], MeshCNN [9], and Ours. From the table, we can see that our performance still outperforms them in average scores.

models. The results are reported in the following table. It is very clearly observed that the performance will decline with more holes or stronger noises.

The experiment of using different numbers of vertices is to demonstrate that our model does not need the correspondences across different shapes. The simplification methods [11,6] are shape-dependent and only related to the geometry features, such as curvatures. They do not guarantee consistent correspondence across different shapes. Moreover, our network is able to be trained with different numbers of vertices, with a batch size of 1. Our model can support an input mesh with an arbitrary number of vertices when testing. We also tested the performance with different vertex numbers in our pre-trained models in the following table.

Indeed, an apple-to-apple comparison is hard. Here, for making a complete evaluation as much as possible, we follow the SubdivNet to evaluate the performance on the vertex segmentation and report the performance on face/edge segmentation on COSEG in the following table. We choose two alternative approaches: PD-MeshNet [17] is evaluated on the faces segmentation, and MeshCNN [9] is evaluated on the edges segmentation. The performance is reported in Table 5, we evaluate the different methods on different metrics (edge, face, and vertex). Ours still outperforms them in average score.

5 Visualization Results on Segmentation

In this section, we provide more visualization results on part segmentation for 16 categories of ShapNet [4] and 3 categories of COSEG [24]. For the 16 categories of ShapeNet, we show 8 examples for each category in Figure 1 and Figure 2. For the COSEG, we display 8 examples for the three largest categories, chairs, vases, and aliens, in Figure 3.

6 Visualization Results on Attention Maps

In this section, we provide more visualization results of attention maps for our proposed dual attention. For each segmented mesh, we illustrate the attention maps from two branches (topology & geometry) as well as their fused module. Each row shows the attention maps for a certain query vertex. More results are shown in Figure 5, where blue to yellow means increasing weights. We can observe that our dual attention is able to determine which part is more important on the specific task, *e.g.* part segmentation. All the displayed shapes are from ShapeNet.

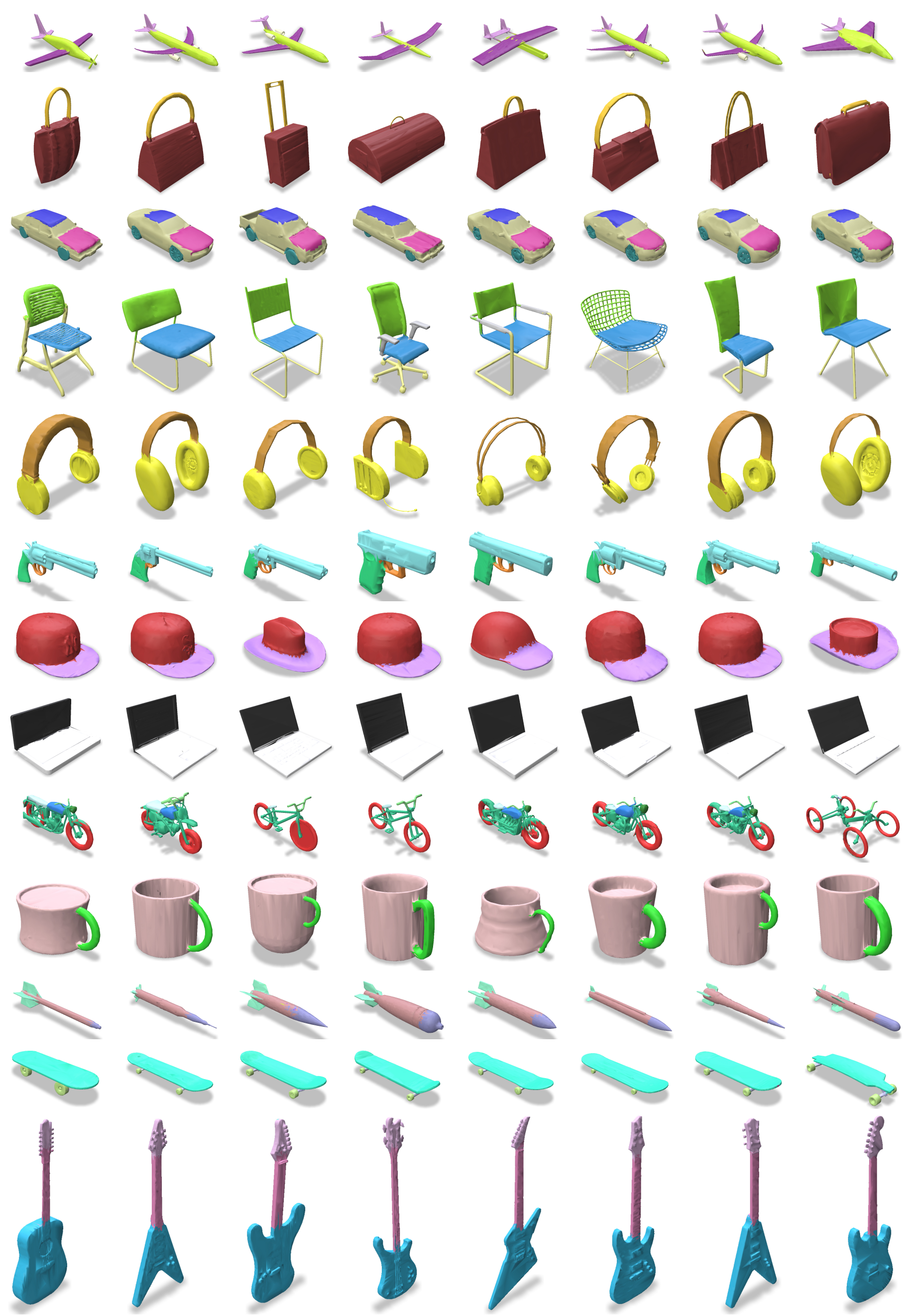

Fig. 1: **Part segmentation results on ShapeNet.** For each category, we display 8 examples.

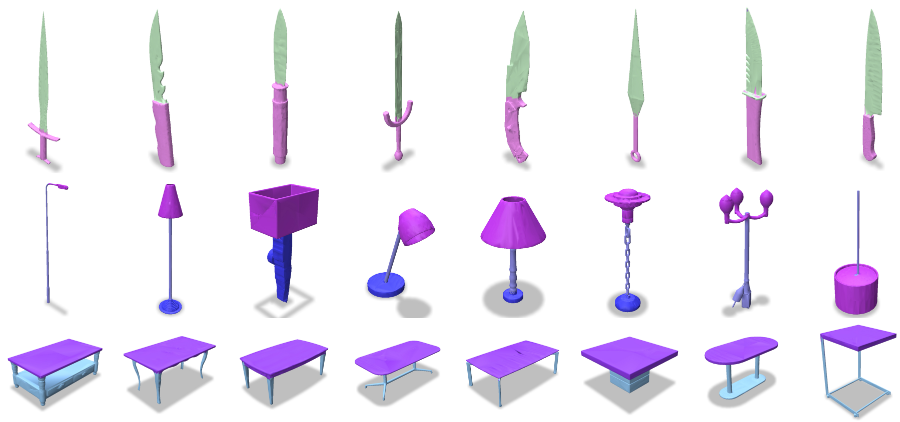

Fig. 2: **Part segmentation results on ShapeNet.** For each category, we display 8 examples.

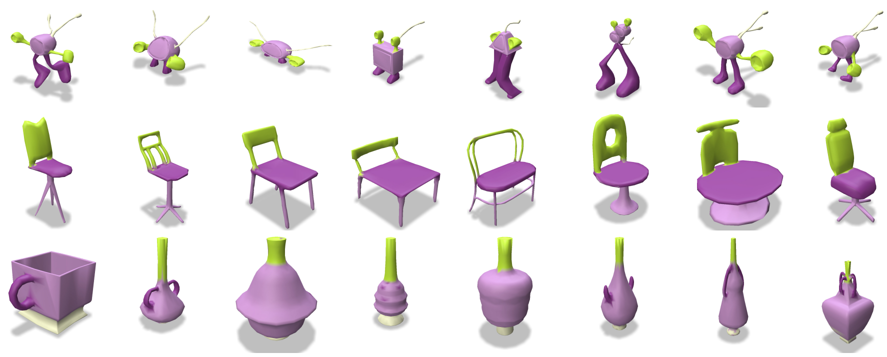

Fig. 3: **Part segmentation results on COSEG datasets.** For each category, we display 8 examples.

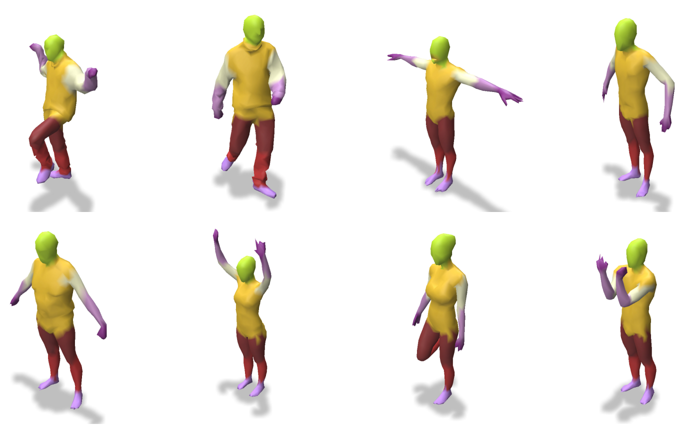

Fig. 4: **Part segmentation results on human body datasets.**

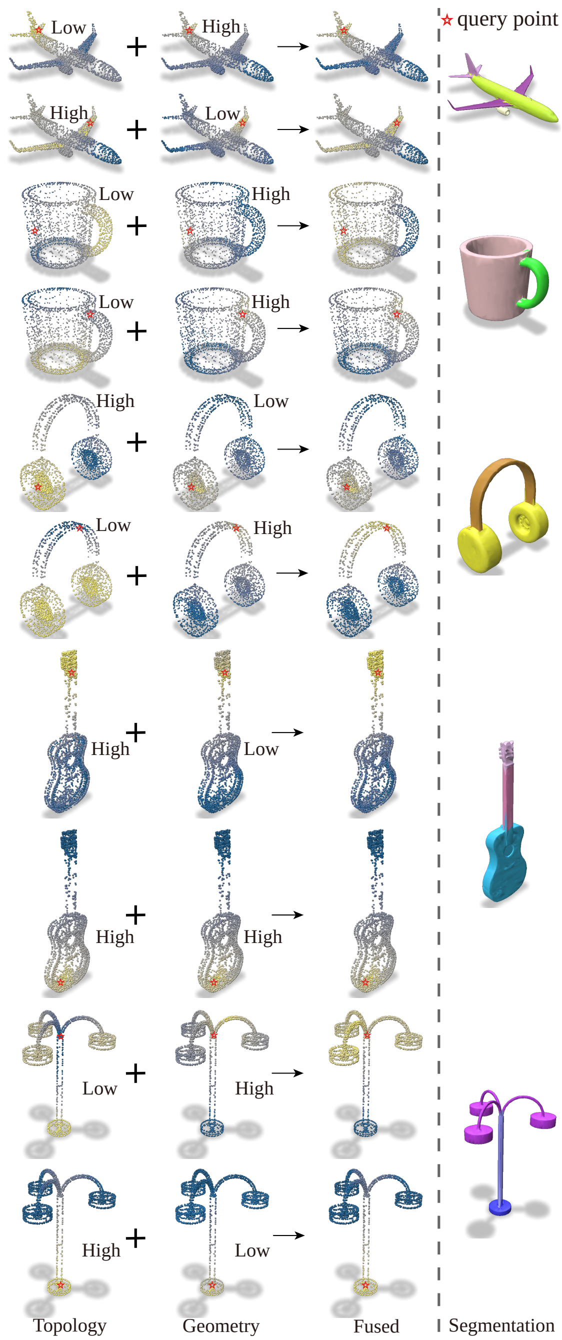

Fig. 5: **Importance Visualization on Attention Maps.** Attention maps of our dual attention mechanism are shown to the left of the segmented shapes.

References

1. Adobe.com: Animate 3d characters for games, film, and more. (2021), https://www.mixamo.com
2. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM SIGGRAPH 2005 Papers, pp. 408–416 (2005)
3. Bogo, F., Romero, J., Loper, M., Black, M.J.: Faust: Dataset and evaluation for 3d mesh registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3794–3801 (2014)
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)
5. Ezuz, D., Solomon, J., Kim, V.G., Ben-Chen, M.: Gwcnn: A metric alignment layer for deep shape analysis. In: Computer Graphics Forum. vol. 36, pp. 49–57. Wiley Online Library (2017)
6. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. pp. 209–216 (1997)
7. Giorgi, D., Biasotti, S., Paraboschi, L.: Shape retrieval contest 2007: Watertight models track. SHREC competition **8**(7) (2007)
8. Haim, N., Segol, N., Ben-Hamu, H., Maron, H., Lipman, Y.: Surface networks via general covers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 632–641 (2019)
9. Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., Cohen-Or, D.: Meshcnn: a network with an edge. ACM Transactions on Graphics (TOG) **38**(4), 1–12 (2019)
10. Hu, S.M., Liu, Z.N., Guo, M.H., Cai, J.X., Huang, J., Mu, T.J., Martin, R.R.: Subdivision-based mesh convolution networks. arXiv preprint arXiv:2106.02285 (2021)
11. Huang, J., Su, H., Guibas, L.: Robust watertight manifold surface generation method for shapenet models. arXiv preprint arXiv:1802.01698 (2018)
12. Lahav, A., Tal, A.: Meshwalker: Deep mesh understanding by random walks. ACM Transactions on Graphics (TOG) **39**(6), 1–13 (2020)
13. Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoua, G., Suetens, P.D., et al.: Shape retrieval on non-rigid 3d watertight meshes. In: Eurographics workshop on 3d object retrieval (3DOR). Citeseer (2011)
14. Liu, H.T.D., Kim, V.G., Chaudhuri, S., Aigerman, N., Jacobson, A.: Neural subdivision. arXiv preprint arXiv:2005.01819 (2020)
15. Maron, H., Galun, M., Aigerman, N., Trope, M., Dym, N., Yumer, E., Kim, V.G., Lipman, Y.: Convolutional neural networks on surfaces via seamless toric covers. ACM Trans. Graph. **36**(4), 71–1 (2017)
16. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 37–45 (2015)
17. Milano, F., Loquercio, A., Rosinol, A., Scaramuzza, D., Carlone, L.: Primal-dual mesh convolutional neural networks. arXiv preprint arXiv:2010.12455 (2020)
18. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)

19. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
20. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace-spectra as fingerprints for shape matching. In: Proceedings of the 2005 ACM symposium on Solid and physical modeling. pp. 101–106 (2005)
21. Sharp, N., Attaiki, S., Crane, K., Ovsjanikov, M.: Diffusionnet: Discretization agnostic learning on surfaces. ACM Transactions on Graphics (TOG) **41**(3), 1–16 (2022)
22. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. In: ACM SIGGRAPH 2008 papers, pp. 1–9 (2008)
23. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
24. Yi, L., Kim, V.G., Ceylan, D., Shen, I.C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.: A scalable active framework for region annotation in 3d shape collections. ACM Transactions on Graphics (ToG) **35**(6), 1–12 (2016)