Learning Delaunay Surface Elements for Mesh Reconstruction
Supplementary Material

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

Paul Guerrero
Adobe Research
guerrero@adobe.com

Noam Aigerman
Adobe Research
aigerman@adobe.com

Niloy Mitra
UCL, Adobe Research
n.mitra@ucl.ac.uk

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

Abstract

In this document we collect additional details about the proposed method and results that were not included in the main manuscript. We evaluate our method on a dataset with non-uniform sampling in Section 1. We demonstrate the generalization power of our method in Section 2 by providing additional quantitative and qualitative results on the ShapeNet dataset. In Section 3 we illustrate our method with more qualitative results. In Section 4 we show example shapes from our dataset. We provide additional details on the architecture of our pipeline in Section 5, show the typical runtime of our method in Section 6, provide ablation of the learned Logmap component in Section 7, and show experiments with different neighborhood sizes in Section 8.

1. Non-uniform sampling

We provide additional results on a non-uniformly sampled variant of the FAMOUSTHINGI dataset. We sample points following a density gradient along the y-axis (horizontal in the figures), where point density correlates with the y-coordinate. A few examples are shown in Figure 1 (bottom). We did *not* retrain on this dataset variant and evaluate the same model we used for the uniform point clouds. In Table 1 we show that our method remains robust even with this non-uniform sampling, with only a small decrease in performance compared to uniform sampling. IER meshing takes the largest performance hit with over twice as many non-manifold triangles and significantly increased Chamfer distance. Overall our method shows a similar improvement over the baselines as in uniform sampling. The angle distribution of triangles produced by our method is compared to all baselines in Figure 2. Our method achieves the best performance with angles more centered around 60 degrees.

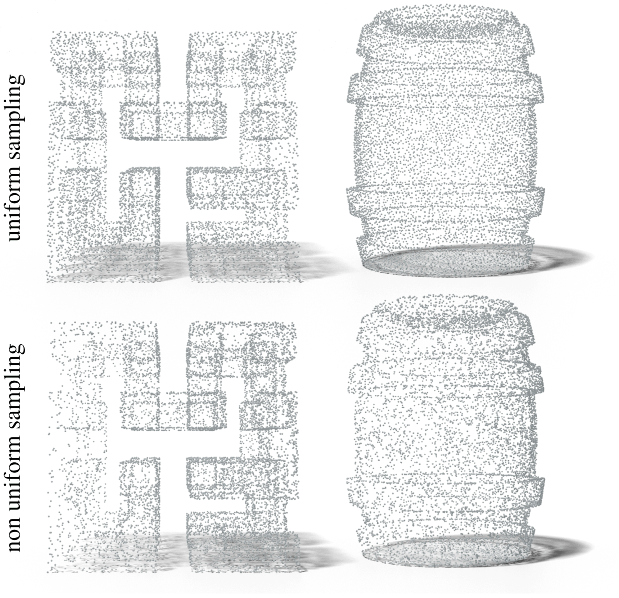

Figure 1. Examples of uniformly sampled point clouds (top) and non uniformly sampled point clouds. The density of points follows a gradient along the y axis (horizontal).

We show qualitative comparison in Figure 3. We observe that ball pivoting and IER meshing are particularly impacted by the non uniform sampling while our method achieves the best quality reconstructions.

Table 1. Quantitative comparison the FamousThingi testset where points are sampled non-uniformly. We compare the percentage of non-watertight edges (NW), the Chamfer distance (CD), and the normal reconstruction in degrees (NR) to all baselines.

Method	NW (%)	CD $*1^{e-2}$	NR
Ball pivoting	31.5	0.396	6.84
PointTriNet	14.2	0.383	6.59
IER meshing	13.5	0.487	7.00
RVE	11.0	0.396	9.08
α-shapes 1%	3.5	3.228	63.21
α-shapes 3%	2.7	0.971	28.88
α-shapes 5%	1.7	1.061	17.71
Ours	**1.3**	**0.356**	**6.02**

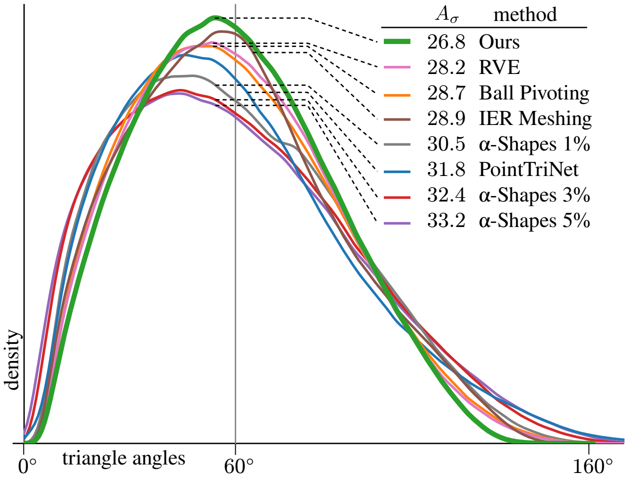

Figure 2. Triangle angles distribution. Our method produces triangles with angles more centered around 60 degrees and fewer very obtuse or very acute angles.

2. Results on ShapeNet

We compare our method to PointTriNet and IER meshing. Both our method and PointTriNet are trained on the FamousThingi dataset, showing their generalization performance, while IER meshing was trained on ShapeNet (since IER meshing requires more shapes for training than the other two methods). Even though this gives IER meshing an advantage, we observe in Table 2 that our method still produces shapes with better manifoldness and Chamfer distance than other methods.

Additional qualitative results are provided in Figure 4. We observe that our method produces meshes with better manifoldness and preserves details such as the drawer handles (row 2) or the two sides of the plane wings (row 1) more accurately. Finally, our method produces fewer large holes in the reconstructed mesh.

Table 2. Quantitative comparison on 100 random shapes from ShapeNet. We compare our three main metrics to learning-based baselines. IER meshing was specifically trained on ShapeNet, while our method trained on a different dataset (FamousThingi). Even with this handicap, our method obtains better manifoldness and Chamfer distance.

Method	NW (%)	CD $*1^{e-2}$	NR
PointTriNet	22.33	0.416	10.95
IER meshing	6.96	0.456	**6.54**
Ours	**5.51**	**0.396**	9.44

3. More Qualitative Results

We provide additional qualitative results by meshing point clouds of well-known monuments obtained from Famous Paris Buildings. Results are shown in Figure 5. Since these shapes are geometrically more complex than the shapes in FamousThingi or ShapeNet, we uniformly sample 50k points from each monument. We do not re-train on this dataset. Our method generalizes well to unseen data and denser point clouds.

We also include a real scan reconstruction in Figure 6. We reconstruct a point cloud with 50k sampled points and compare to other learning-based methods. Since IER meshing can not handle 50k points, we sample 12k points for comaring to IER meshing.

4. Dataset Examples

A few examples of shapes from our FamousThingi dataset are shown Figure 7. In Figure 1, we show two examples of uniformly sampled point clouds we use as input to our method, and two non-uniformly sampled point clouds that we use in the experiments described in Section 1.

5. Architecture Details

We show the detailed architecture of our geodesic patch *classification network* and the 2D log map *projection network* in Figure 8. The classification network implements a function $c_j := f_\theta([q^i_j, d^i_j] \mid Q^i)$ that classifies if each point q_i in the euclidean patch Q^i is part of the geodesic patch P_i, while the projection network implements a function $u^i_j := g_\phi([p^i_j, d^i_j] \mid P^i)$ that projects points p^i_j in the geodesic patch P_i to their 2D log map coordinates u^i_j. Here d_i is the euclidean distance from a point to its patch center.

Both networks use the same architecture based on FoldingNet [4], except for their output dimension. They take as input a 3D point concatenated with the distance to the patch center and proceed to compute a 1024-dimensional global feature vector for the input patch with a PointNet [3]. Each input point is then augmented with this global feature vector and transformed by two blocks of per-point MLPs into a one-dimensional (classification network) or two-dimensional (projection network) per-point output.

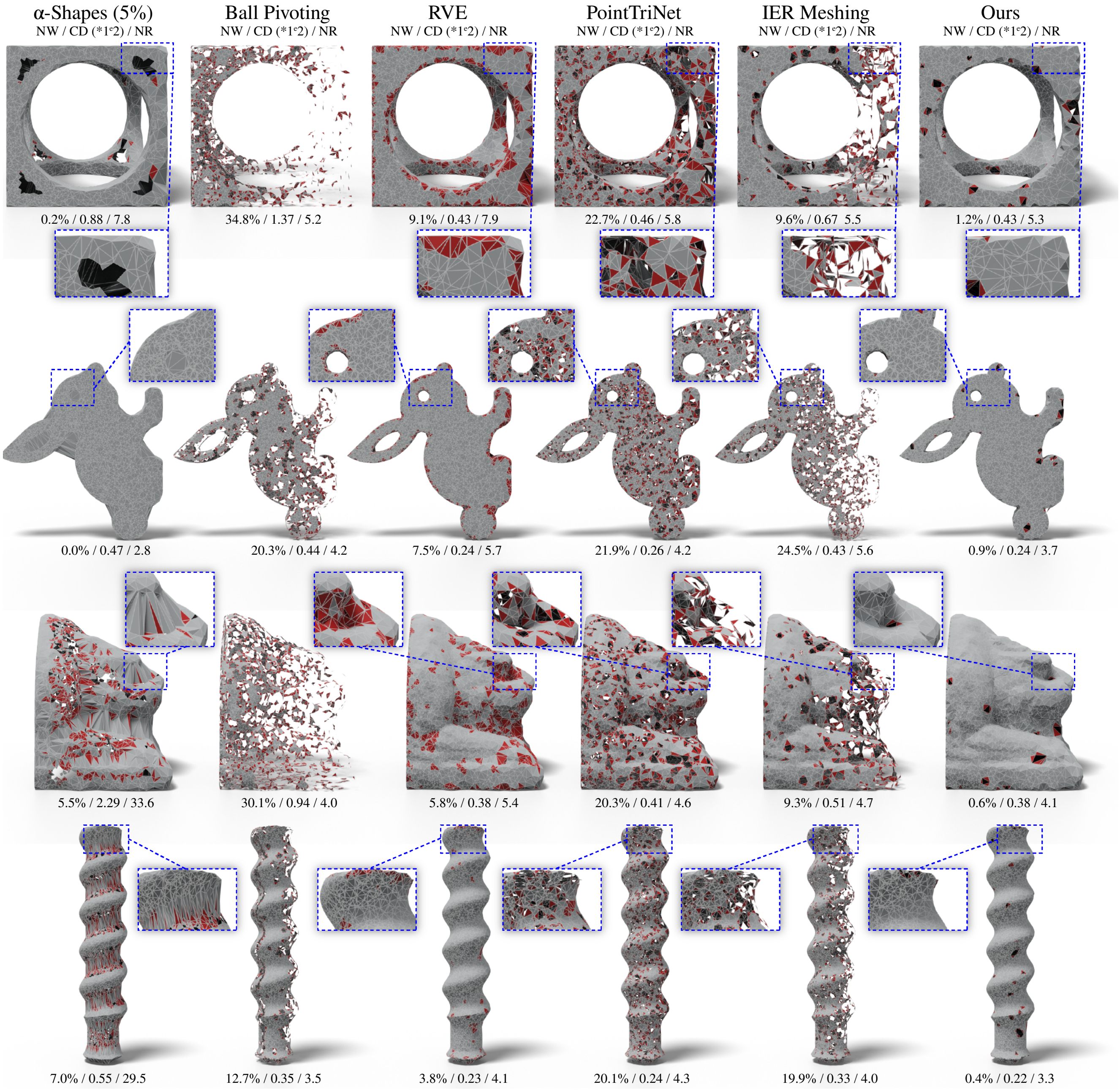

Figure 3. Surface reconstructions from non-uniform point clouds. Non-manifold triangles are marked in red. Shapes are sampled more densely to the left and more coarsely to the right. We can see that methods struggle to reconstruct the coarsely sampled parts of the point cloud. While our method also has slightly more errors in the coarsely sampled regions, the mesh quality drops by a much smaller amount from densely to coarsely sampled regions.

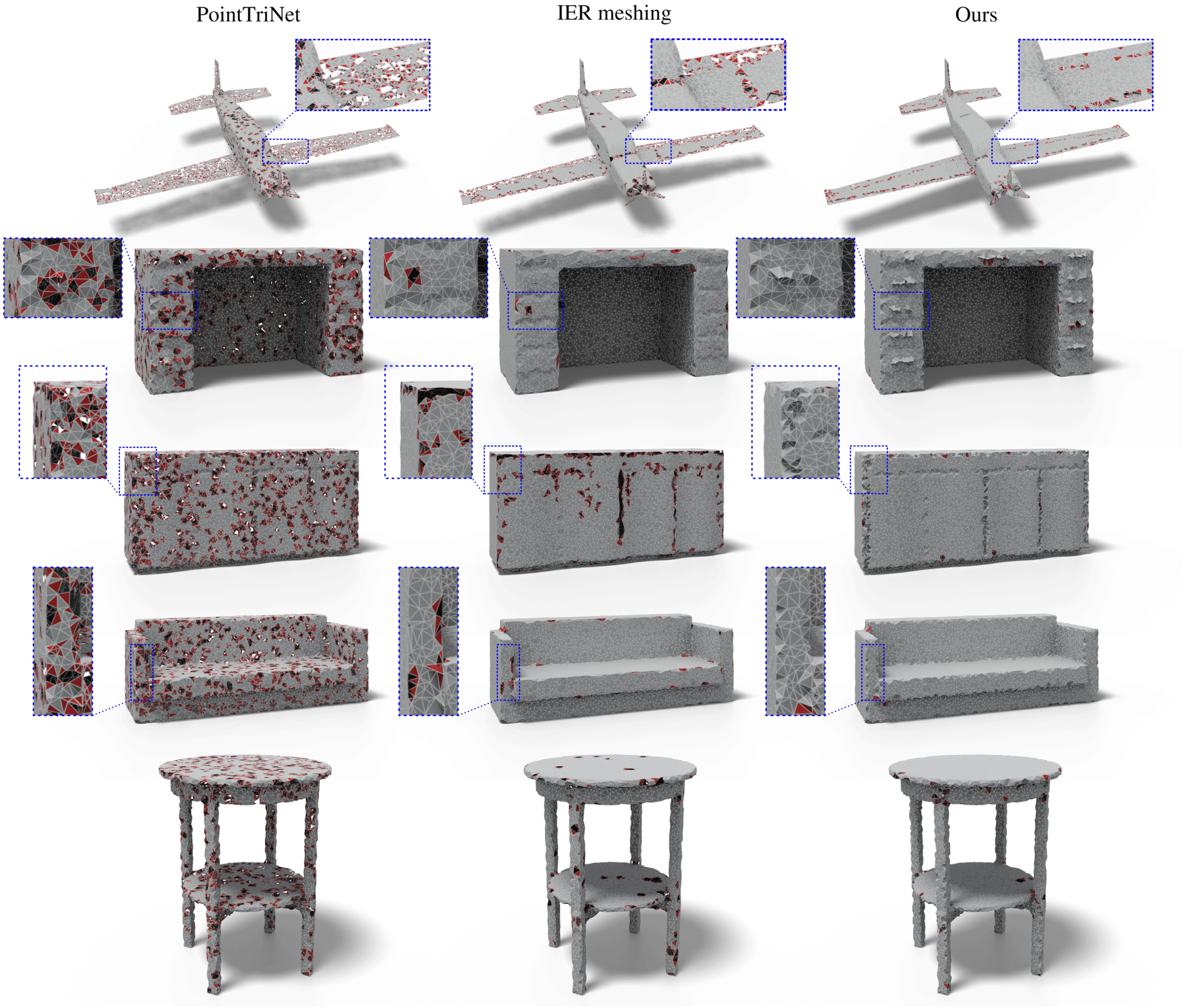

Figure 4. Qualitative results on ShapeNet testset. We do not retrain our method on the ShapeNet dataset while IER meshing *was* trained on this dataset. Even so, our method produces more manifold meshes and preserves details such as the drawer handles (row 2) more accurately. We better separate the two sides (top and bottom) of the plane wings (row 1). Finally, our method presents fewer large holes in the reconstructed mesh.

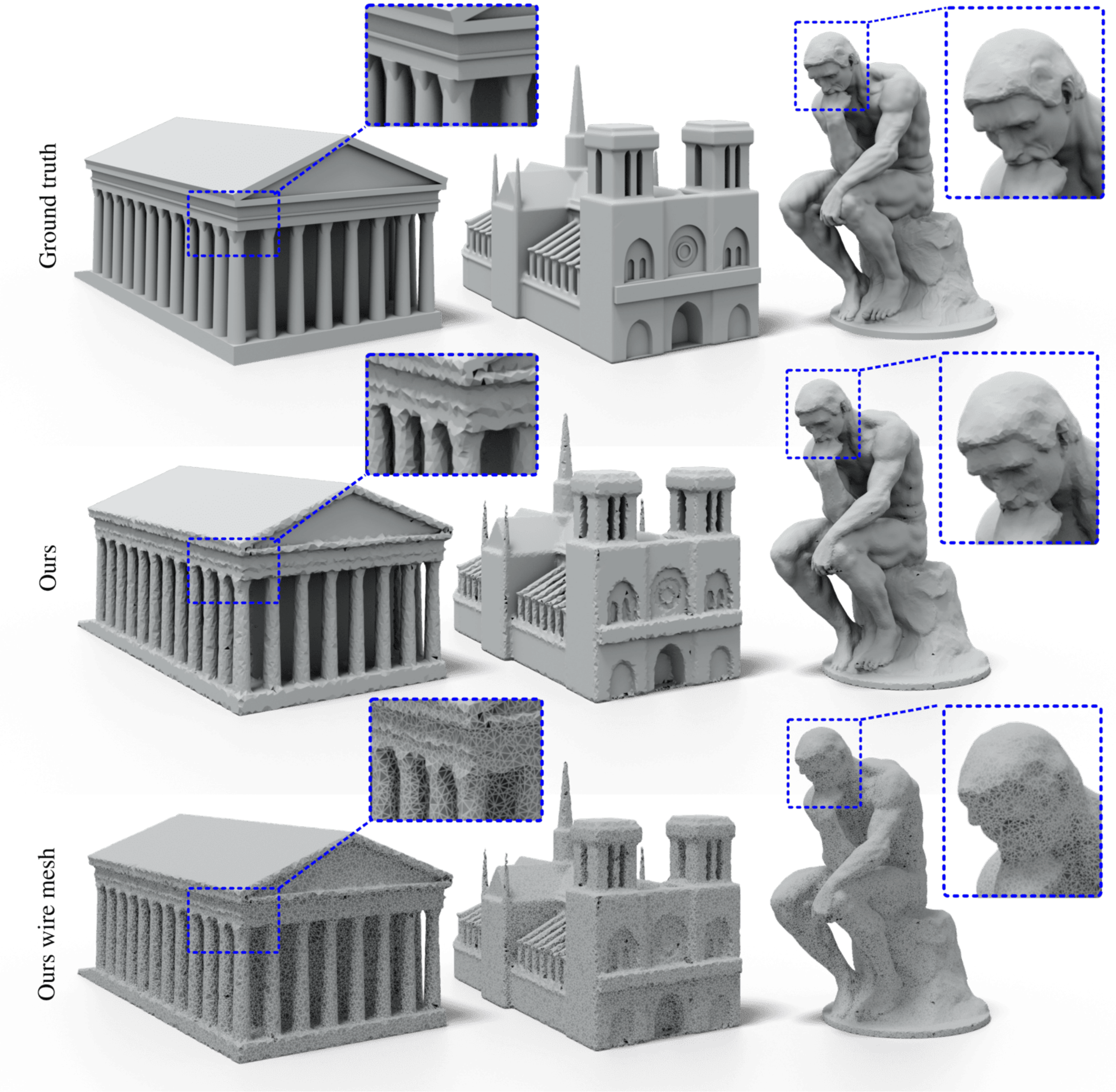

Figure 5. Meshing well-known monuments. We show the ground truth (top), the reconstructed mesh (middle), the reconstructed mesh with non manifold triangles colored in red (bottom). Our method generalizes well this more complex data that is also sampled more densely than our training set.

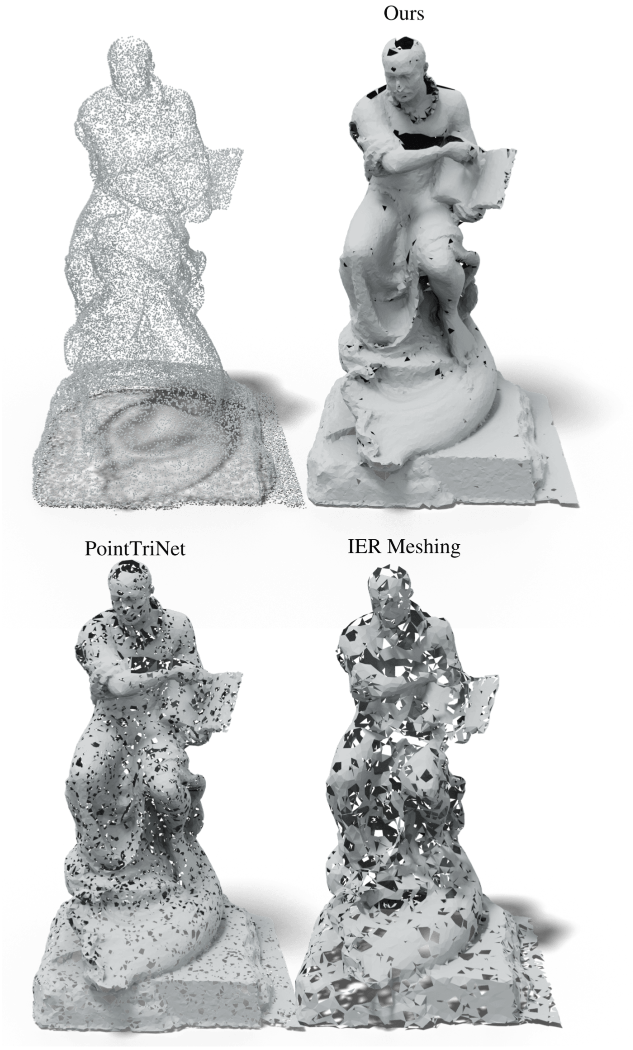

Figure 6. Reconstructing real scans from Tanks and Temples [2]

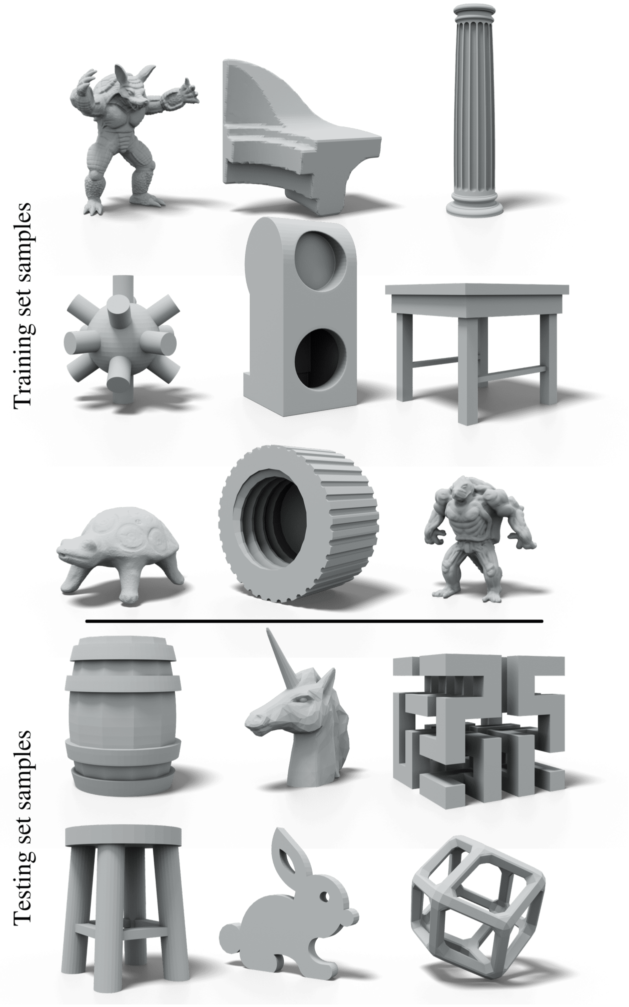

Figure 7. Examples from our dataset. We show ground truth meshes from both the training set and the test set.

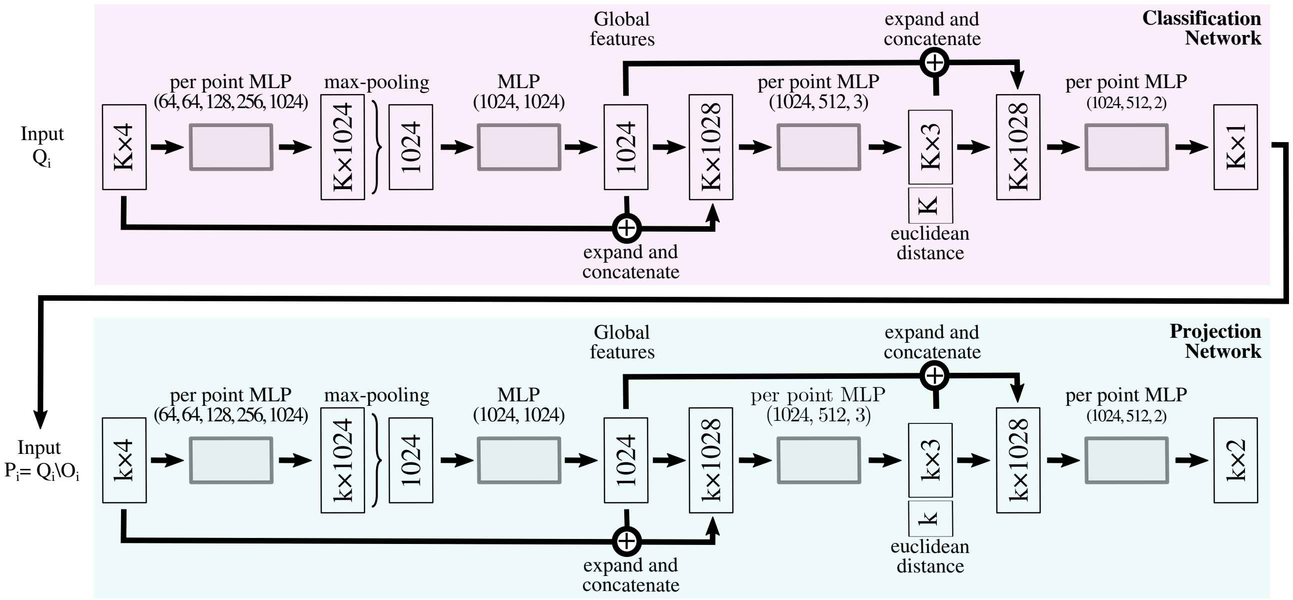

Figure 8. Detailed architecture of our pipeline. We first select a small geodesic patch using the classification network (purple). The projection network (blue) then applies a 2D projection to this patch that approximates a log map.

Table 3. Average runtime estimation per step on 10k point clouds in seconds.

Log map est.	Log map align.	Selection	Total
5.8	24.8	2.1	32.8

Table 4. Quantitative comparison our our learned logmap component to two logmap approximation methods.

	geodesic distance $*1^{e-3}$	2D position $*1^{e-2}$
Projection	1.943	2.627
Rotation	1.943	2.835
Ours	**0.471**	**0.835**

6. Runtime

We measure the average runtime of our method on point clouds of 10k vertices in Table 3, including the runtime for each step of the method.

7. Ablation of the Learned Logmap

We compare the performance of our learned logmap component to two baselines. The first baseline approximates the logmap by projecting neighboring points onto the tangent plane computed from the ground truth normal. The second baseline is the approach proposed in [1], where points are rotated onto the tangent plane. Please note the both of these baselines use ground truth normal information, while our method does not. We evaluate the methods on a subset of 33 manifold shapes from our FAMOUSTHINGI testset and sample 2k patches of k=30 geodesic neighbors per shape. We measure the MSE of the geodesic distance and of the 2D coordinates after patch alignment. Our method produces significantly better logmap estimates compared to other baselines as we show in Table 4.

8. Ablation of Neighbor Counts k and K

We evaluate our method on different values of the geodesic neighbor count k (20, 30, and 50) and different values for the euclidean neighbor count K (80, 120, 160) in Table 5. For each pair of (k, K) values, we train our models for 30 epochs. The choice of the geodesic neighbor count k affects the performance of our method significantly. If k is small, the Delaunay element approximation quality is affected. If k is large, it is more difficult for the logmap estimation network to produce a usable logmap. Changes in the choice of the euclidean neighbor count K lead to less significant performance drops. In our experiment we choose the parameter values $k = 30$ and $K = 120$ which produce the best results for the non-watertightness and normal reconstruction metrics. Please note that for $k = 30$ the difference in Chamfer distance values is negligible.

Table 5. Ablation of different values for the geodesic and euclidean and neighbor counts k and K.

k	K	CD($*1^{e-2}$)	NW(%)	NR
20	80	0.3437	5.569	5.921
20	120	0.3394	4.381	5.845
20	160	0.3496	4.712	6.483
30	80	0.3274	0.509	5.682
30	120	0.3276	**0.485**	**5.661**
30	160	**0.3272**	0.524	5.690
50	80	0.3335	1.822	6.046
50	120	0.3282	0.667	5.856
50	160	0.3286	0.728	5.883

References

[1] Meenakshisundaram Gopi, Shankar Krishnan, and Cláudio T Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. In *Computer Graphics Forum*, volume 19, pages 467–478. Wiley Online Library, 2000. 8

[2] Arno Knapitsch et al. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM TOG*, 36(4), 2017. 6

[3] 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

[4] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. 2