SUPPLEMENTARY MATERIAL

Huan Lei, Ruitao Leng, Liang Zheng, Hongdong Li
School of Computing, The Australian National University

Implementation details. We use $\eta_0 = 0.01$ to adjust the KNN patch resolution. To obtain the anchor priors, we set the maximum radius as $R = 0.2$ (i.e. $20 \times \eta_0$), and the splitting steps along different axes as $\Delta\rho = 0.02$ (i.e. $2 \times \eta_0$), $\Delta\theta = \frac{\pi}{6}, \Delta\phi = \frac{\pi}{6}$. It results in a total number of 720 anchor points. We apply a multiplier of $\beta = 16$ for the graph convolution, a level of $L = 16$ for the positional encoding, and a ratio of 1:20 between positive and negative samples in Eq. (8). We set $\lambda = 1$ in the final loss computation. To train the network, we sample KNN patches with $K = 50$ randomly from the point cloud. Standard geometric transformations such as scaling, rotating, noisy perturbation, are applied to augment the point cloud. We use Adam optimizer (Kingma & Ba 2015) with an initial learning rate of 0.001, which is decayed by 0.5 per 8×10^4 iterations. Our batch size for the KNN patches is 400. The number of training parameters in the proposed CircNet is 7.5 million (7.5M). See Fig. A for the training details of our detection network. To demonstrate the quality of our local predictions/triangulations, we report additional metrics in terms of the average accuracy (mAcc) and average intersection-over-union (mIoU) of the KNN patches. Per triangle forms a prediction element in those computations.

Surface quality. We evaluate the overall quality of each reconstructed mesh using Chamfer Distance (CD1), squared Chamfer (CD2) and F-Score (F1), and the quality of their surface normals with Normal Consistency (NC) and Normal Reconstruction error in degrees (NR). Preserving sharp edges is also important for the surface reconstruction. We therefore evaluate the Edge Chamfer Distance (ECD1) and Edge F-score (EF1) of the reconstructed meshes. Note that the computations of metrics CD1, CD2 and F1 require points to be sampled densely on the entire surface, while those of ECD1 and EF1 utilizes only points sampled near the edges and corners. We follow the convention (Sharp & Ovsjanikov 2020) to sample 10^5 points on the ground-truth and reconstructed meshes for evaluating the quality of shapes. While for the evaluation of large-scale scenes in Matterport3D, one million points are sampled for better coverage of the surface.

For explicity, we provide the definitions of Chamfer distance and F-score below. Given a ground-truth mesh $\mathcal{T}$ and its reconstructed mesh $\widehat{\mathcal{T}}$, we sample the same number of points (*e.g.*, 10^5) uniformly on each of them. Let the resulted point clouds be $\mathcal{Q}$ and $\widehat{\mathcal{Q}}$, respectively. The chamfer distance (CD1) is calculated as

$$\text{CD1}(\mathcal{Q}, \widehat{\mathcal{Q}}) = \frac{1}{|\mathcal{Q}|} \sum_{\mathbf{x} \in \mathcal{Q}} \min_{\mathbf{y} \in \widehat{\mathcal{Q}}} \|\mathbf{x} - \mathbf{y}\|_2 + \frac{1}{|\widehat{\mathcal{Q}}|} \sum_{\mathbf{x} \in \widehat{\mathcal{Q}}} \min_{\mathbf{y} \in \mathcal{Q}} \|\mathbf{x} - \mathbf{y}\|_2. \tag{1}$$

The first term measures completeness of the reconstructed mesh, while the second term measures its accuracy. For the squared chamfer distance (CD2), $\|\mathbf{x} - \mathbf{y}\|_2$ is replaced as $\|\mathbf{x} - \mathbf{y}\|_2^2$. The computation of F-score follows its standard definition, *i.e.*,

$$\text{F1} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \tag{2}$$

The recall and precision are calculated as

$$\text{recall} = \frac{1}{|\mathcal{Q}|} \sum_{\mathbf{x} \in \mathcal{Q}} \mathbf{1}\big((\min_{\mathbf{y} \in \widehat{\mathcal{Q}}} \|\mathbf{x} - \mathbf{y}\|_2) < \epsilon\big), \tag{3}$$

$$\text{precision} = \frac{1}{|\widehat{\mathcal{Q}}|} \sum_{\mathbf{x} \in \widehat{\mathcal{Q}}} \mathbf{1}\big((\min_{\mathbf{y} \in \mathcal{Q}} \|\mathbf{x} - \mathbf{y}\|_2) < \epsilon\big), \tag{4}$$

where ϵ is small threshold for the distance and $\mathbf{1}(\cdot)$ is the indicator function.

Efficiency. In addition to surface quality, a triangulation method is expected to reconstruct high-quality mesh in short time. For efficiency comparison, we report the total reconstruction time of each method on the same machine with one NVIDIA GeForce RTX 2080Ti GPU and AMD Ryzen

Threadripper 2990WX CPU. The results of alpha-shapes and ball-pivot are computed based on the latest 'pymeshlab' (i.e. 2022.2.post2) for Python. To be more specific, suppose the bounding box diagonal of a point cloud is l_D. We report the performance of α-shapes using $\alpha = 3\% \times l_D$ and $\alpha = 5\% \times l_D$. For ball-pivoting, the radius is set to be $1\% \times l_D$. The results of PSR are produced using 'open3d' (i.e. 0.15.2+c074f5d) by setting depth to 9.

More results on robustness. We show the quantitative comparisons of different learning-based triangulation methods on the ABC point clouds that are uniform, different levels of noise, and non-uniform distributions in Table A, Table B, Table C, Table D, Table E, respectively. We also report their performance on data that are sampled under random uniformity in Table F.

More visualizations. Figure B shows the reconstructed meshes of different methods for another building in Matterport3D. The proposed CircNet consistently produces mesh of high quality. We also compare the reconstruction quality of different methods, using complex shapes in the Thingi10K dataset (Zhou & Jacobson 2016). It can be seen from Fig. C that the learning-based triangulation methods, *i.e.*, DSE, PointTriNet and the proposed CircNet are all robust to noise. For the non-uniform point cloud, only PointTriNet and the proposed CircNet reconstruct the complete underlying shape without chopping out the 8 thin tails. The other methods either reconstruct an incomplete mesh or overcomplete mesh. The number right below each mesh indicates the percentage of non-manifold edges of each specific mesh. Figure D shows a further example.

Hyperparameters. We evaluate the effect of neighborhood size (K) and the number of predictions per anchor cell (s) on our model performance. Specifically, we compare the model performance of $K = 50$ to that of $K \in \{25, 100, 200\}$. It can be seen from Table G that $K = 100$ and $K = 200$ perform similarly to $K = 50$, while $K = 25$ results in performance drop. The reason is that $K = 25$ is too small to cover the 1-ring neighborhood. Given similar performance, smaller K is preferred for better efficiency. As for s, we do not recommend settings of $s > 2$ because dominantly, each anchor cell is observed to be populated by at most two ground-truth circumcenters. We hence compare the model performance of $s = 2$ to that of $s = 1$ only. It shows that $s = 2$ performs slightly better on the overall reconstruction quality.

REFERENCES

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Nicholas Sharp and Maks Ovsjanikov. Pointtrinet: Learned triangulation of 3d point sets. In *European Conference on Computer Vision*, pp. 762–778. Springer, 2020.

Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

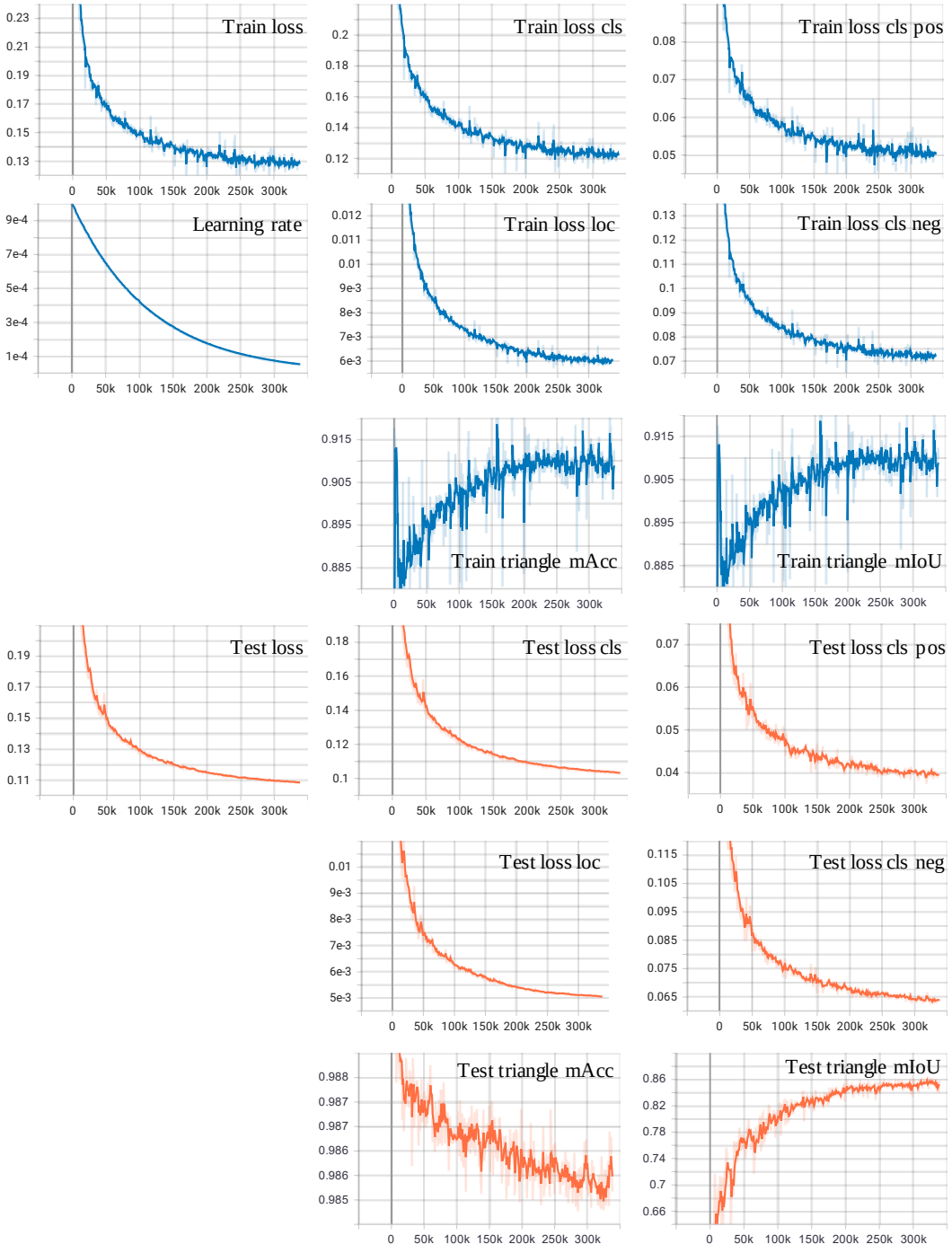

Figure A: Training and test curves. The mAcc and mIoU metrics are computed based on per triangle.

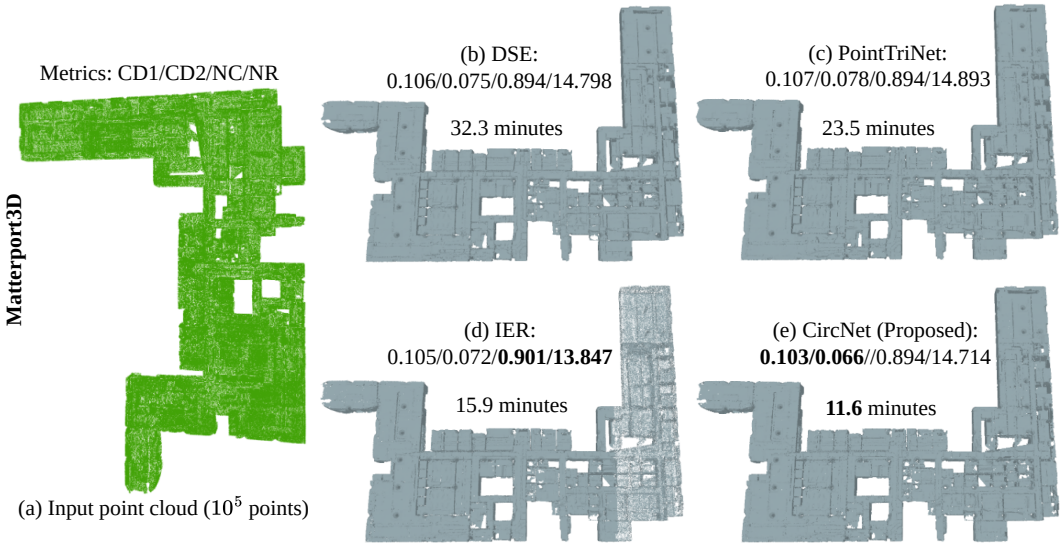

Figure B: Reconstructed scene meshes of different learning-based triangulation methods. We compare their surface quality using the metrics CD1, CD2, NC, and NR. It can be seen that the mesh of CircNet has the lowest Chamfer distances.

Table A: Poisson.

Method	Surface Quality						
	overall					sharp	
	CD1($\times 10^2$)↓	CD2($\times 10^5$)↓	F1↑	NC↑	NR↓	ECD1($\times 10^2$)↓	EF1↑
α-shapes-3%	0.421	2.460	0.853	0.948	7.153	4.067	0.434
α-shapes-5%	0.536	5.844	0.834	0.942	7.650	4.462	0.442
ball-pivot (+$\mathbf{n}$)	0.286	0.592	0.938	0.975	4.279	1.514	0.607
DSE	0.276	0.524	0.952	0.981	3.560	1.124	0.667
IER	0.282	0.572	0.945	0.981	3.300	1.065	0.718
PointTriNet	0.278	0.532	0.950	0.978	3.921	1.372	0.667
CircNet (Prop.)	0.278	0.533	0.949	0.976	4.033	1.281	0.660

Table B: Noise level: $\sigma = 0.1$.

Method	Surface Quality						
	overall					sharp	
	CD1($\times 10^2$)↓	CD2($\times 10^5$)↓	F1↑	NC↑	NR↓	ECD1($\times 10^2$)↓	EF1↑
α-shapes-3%	0.453	2.501	0.840	0.948	9.697	4.579	0.398
α-shapes-5%	0.564	5.746	0.822	0.943	9.465	5.301	0.397
ball-pivot (+$\mathbf{n}$)	0.339	0.807	0.892	0.963	10.260	2.392	0.544
DSE	0.326	0.711	0.910	0.970	9.541	1.187	0.639
IER	0.348	0.849	0.881	0.967	9.242	1.413	0.655
PointTriNet	0.328	0.718	0.908	0.966	9.899	1.581	0.634
CircNet (Prop.)	0.328	0.720	0.908	0.965	10.053	1.500	0.630

Table C: Noise level: $\sigma = 0.2$.

Method	Surface Quality						
	overall					sharp	
	CD1($\times 10^2$)↓	CD2($\times 10^5$)↓	F1↑	NC↑	NR↓	ECD1($\times 10^2$)↓	EF1↑
α-shapes-3%	0.542	3.031	0.745	0.942	11.667	6.034	0.315
α-shapes-5%	0.676	6.896	0.716	0.938	11.136	6.918	0.300
ball-pivot (+$\mathbf{n}$)	0.429	1.347	0.783	0.931	16.574	4.901	0.390
DSE	0.415	1.226	0.797	0.937	16.234	2.695	0.515
IER	0.446	1.439	0.761	0.945	13.702	3.327	0.500
PointTriNet	0.416	1.234	0.795	0.934	16.430	3.519	0.500
CircNet (Prop.)	0.419	1.245	0.793	0.931	16.735	4.396	0.467

Table D: Noise level: $\sigma = 0.3$.

Method	Surface Quality						
	overall					sharp	
	CD1($\times 10^2$)↓	CD2($\times 10^5$)↓	F1↑	NC↑	NR↓	ECD1($\times 10^2$)↓	EF1↑
α-shapes-3%	0.653	3.965	0.622	0.934	13.471	6.764	0.260
α-shapes-5%	0.798	8.036	0.562	0.932	12.529	7.402	0.245
ball-pivot (+$\mathbf{n}$)	0.524	2.127	0.697	0.898	21.351	6.819	0.293
DSE	0.517	2.033	0.697	0.883	23.386	6.311	0.325
IER	0.545	2.235	0.668	0.929	16.030	5.682	0.340
PointTriNet	0.517	2.038	0.698	0.888	22.738	6.353	0.324
CircNet (Prop.)	0.523	2.076	0.689	0.880	23.287	6.790	0.294

Table E: Non-uniform data.

Method	Surface Quality						
	overall					sharp	
	CD1($\times 10^2$)↓	CD2($\times 10^5$)↓	F1↑	NC↑	NR↓	ECD1($\times 10^2$)↓	EF1↑
α-shapes-3%	0.475	3.407	0.825	0.936	8.875	4.760	0.367
α-shapes-5%	0.543	5.723	0.820	0.939	8.308	4.987	0.383
ball-pivot (+$\mathbf{n}$)	1.237	43.581	0.653	0.929	9.288	8.538	0.418
DSE	0.353	1.325	0.891	0.955	7.238	2.842	0.495
IER	0.542	6.629	0.833	0.960	6.044	4.980	0.547
PointTriNet	0.342	1.103	0.894	0.956	7.357	2.535	0.514
CircNet (Prop.)	0.395	1.852	0.858	0.942	8.693	3.490	0.457

Table F: Random.

Method	Surface Quality						
	overall					sharp	
	CD1($\times 10^2$)↓	CD2($\times 10^5$)↓	F1↑	NC↑	NR↓	ECD1($\times 10^2$)↓	EF1↑
α-shapes-3%	0.424	2.514	0.851	0.946	7.358	4.248	0.380
α-shapes-5%	0.533	5.750	0.834	0.942	7.634	4.969	0.392
ball-pivot (+$\mathbf{n}$)	0.356	1.327	0.871	0.967	5.471	1.987	0.554
PointTriNet	0.284	0.565	0.942	0.973	4.752	1.863	0.604
IER	0.289	0.599	0.940	0.974	4.475	1.433	0.640
DSE	0.282	0.555	0.945	0.976	4.254	1.380	0.601
CircNet (Prop.)	0.316	0.815	0.905	0.963	5.699	2.373	0.553

Table G: Effects of different hyperparameters on the performance.

Method	Surface Quality						
	overall					sharp	
	CD1($\times 10^2$)↓	CD2($\times 10^5$)↓	F1↑	NC↑	NR↓	ECD1($\times 10^2$)↓	EF1↑
$k = 25, s = 2$	0.285	0.548	0.949	0.983	2.012	0.786	0.917
$k = 50, s = 1$	0.285	0.636	0.950	0.985	1.768	**0.686**	0.925
$k = 50, s = 2$	0.284	0.545	0.950	0.985	1.778	0.712	0.924
$k = 100, s = 2$	**0.284**	**0.544**	**0.950**	**0.985**	**1.766**	0.691	**0.925**
$k = 200, s = 2$	0.284	0.545	0.950	0.985	1.776	0.697	0.925

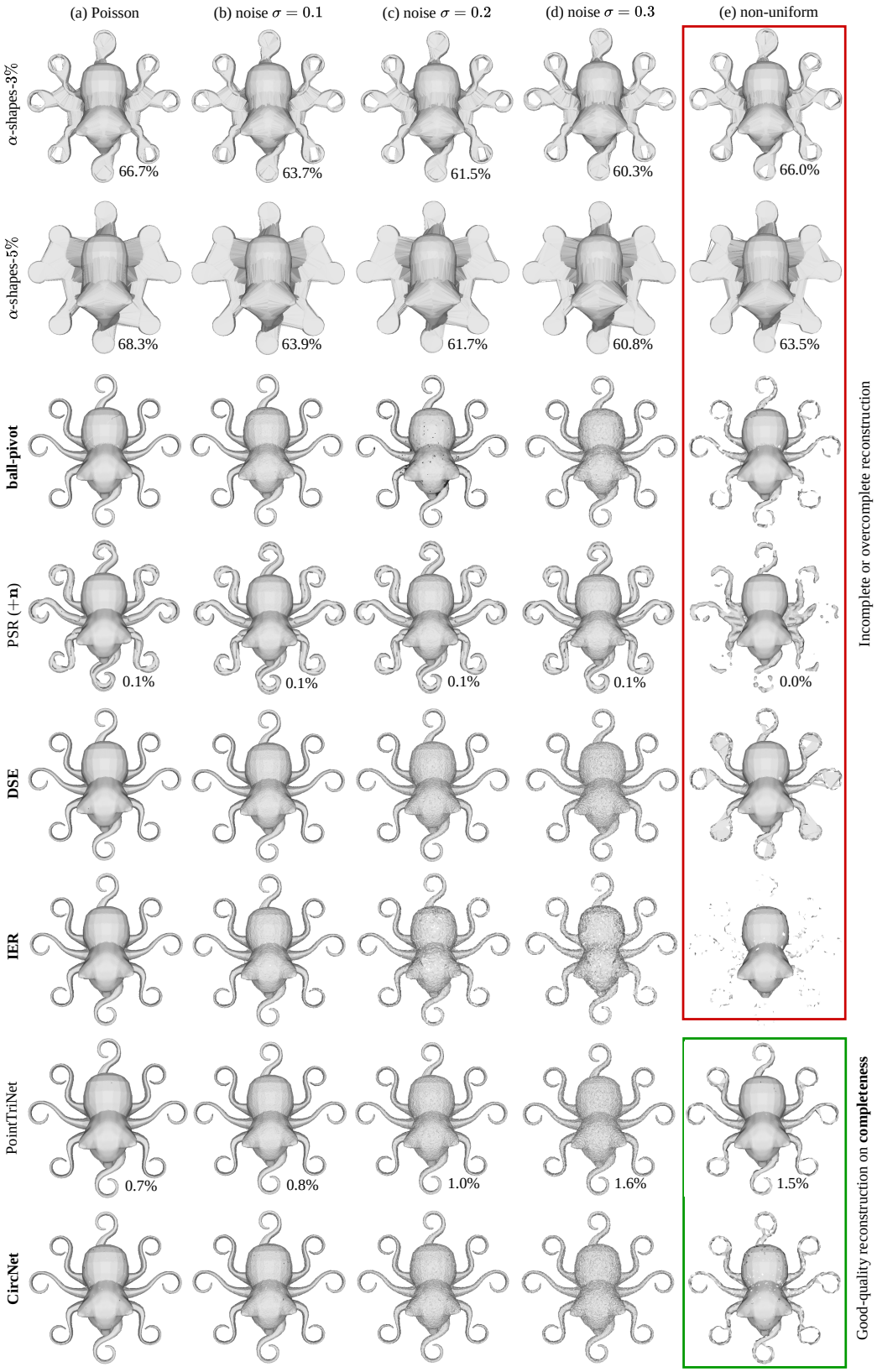

Figure C: Reconstructed surface shapes of different methods for noisy and non-uniform point clouds. Among those methods, ball-pivot, DSE, IER and the proposed CircNet guarantee an edge-manifold surface as output, while the other methods produce non-manifold meshes. We show the percentage of non-manifold edges of each reconstructed mesh right below it.

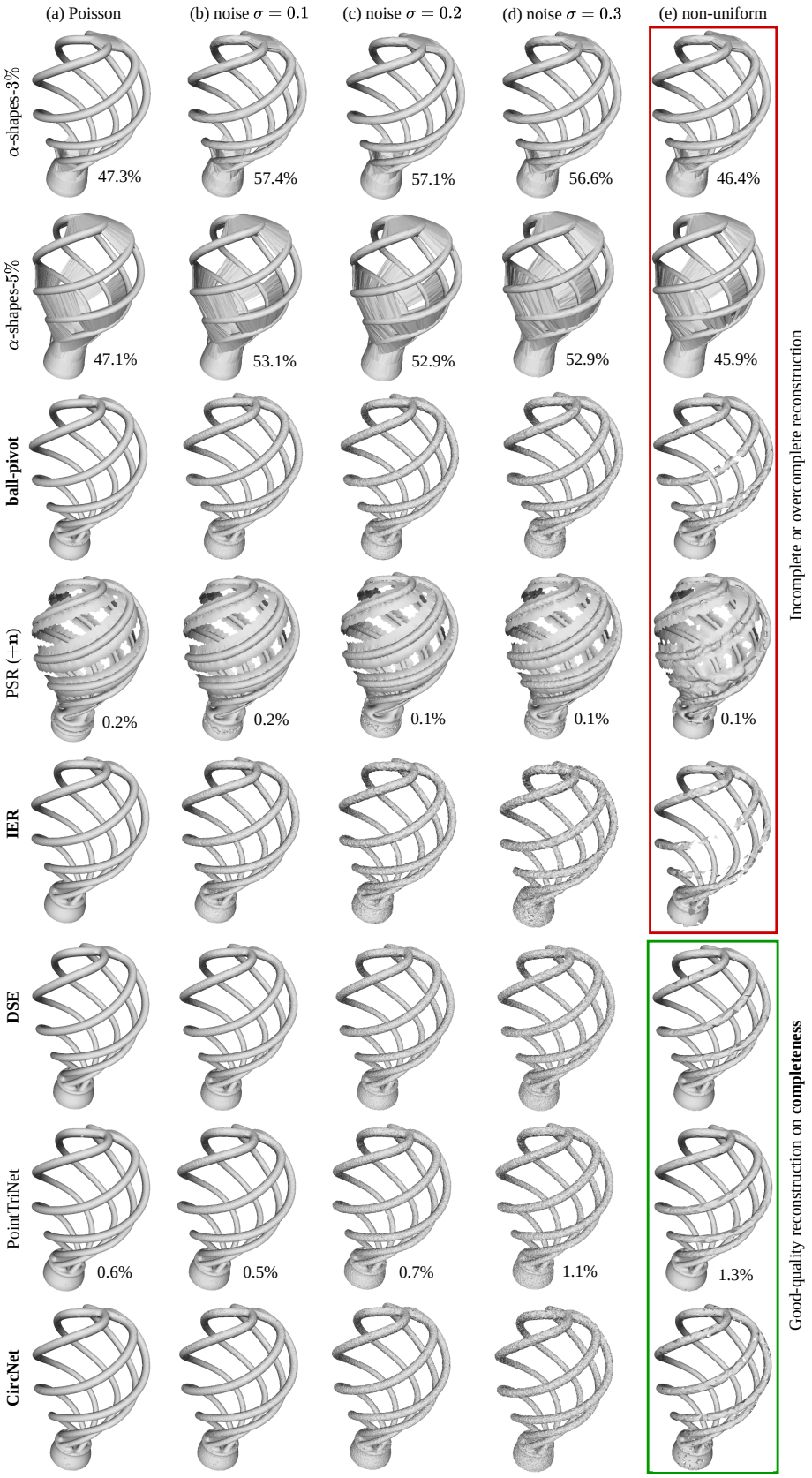

Figure D: Further examples of the reconstructed surface shapes of different methods for noisy and non-uniform point clouds.