Rethinking the Approximation Error in 3D Surface Fitting for Point Cloud Normal Estimation: Supplementary Materials

Hang Du*, Xuejun Yan*, Jingjing Wang, Di Xie†, and Shiliang Pu
Hikvision Research Institute, Hangzhou, China
{duhang, yanxuejun, wangjingjing9, xiedi, pushiliang.hri}@hikvision.com

In the supplementary materials, we provide implementation details of network architecture. For a thorough evaluation, we conduct more experiments, including ablation studies on the z-direction transformation loss, robustness against neighborhood sizes and Jet orders n, iterative estimation experiment and more visualization results. In addition, we also provide an application of normal estimation, *i.e*. surface reconstruction, to further verify the effectiveness of our methods.

1. Network Architecture Details

In this section, we provide the details of network architecture.

1.1. GCN-based Transformation Network

In order to accomplish z-direction transformation, we design a GCN-based spatial transformation network, which is shown in Fig. 1. In particular, EdgeConv (3, 64) denotes a EdgeConv [5] layer with the number of input channel as 3, and the number of output channel as 64. AdaGP (64, 128) represents an adaptive graph pooling [6] layer with the number of input/output channel as 64, and the output number of point as 128. Conv1d (256, 512) indicates a 1D convolutional layer with the number of input channel as 256, and the number of output channel as 512. Avg. and Max pooling denotes a combination of average and max pooling operations. FC (1024, 512) indicates a fully-connected layer with the number of input channel as 1024, and the number of output channel as 512.

1.2. Normal Error Estimation

The network architecture of the normal error estimation is shown in Fig. 2. Specifically, the point-wise feature is fed into a 1D convolutional layer with the number of input/output channel as 128. Subsequently, through a max pooling layer, the global feature is concatenated with the rough estimated normal to estimate the normal error. Then,

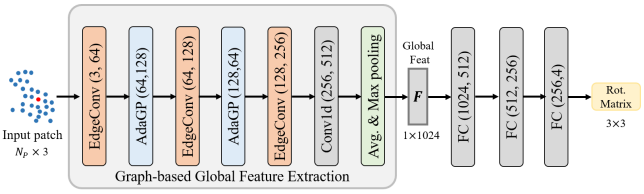

Figure 1. Detailed architecture of GCN-based spatial transformation network.

the error of normal estimation is added on the rough estimation to obtain the output normal. Finally, the output normal is normalized into a unit vector.

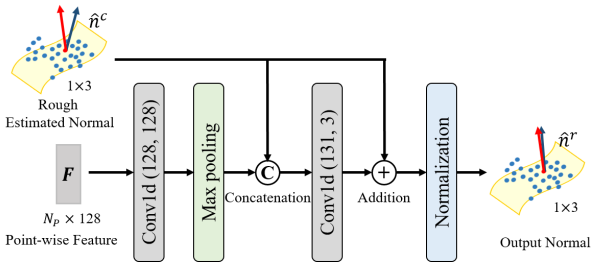

Figure 2. Architecture details of the normal error estimation network.

2. More Experimental Results

In this section, we first conduct more ablation studies on the proposed methods. Then, we provide an application of normal estimation to surface reconstruction. Finally, we give more visualization results.

2.1. Ablation Study

Robustness to the neighborhood size. In the main text, we have reported the results of DeepFit [1], AdaFit [7], and GraphFit [3] under their optimal input neighborhood sizes. To further verify the robustness to the neighborhood sizes, we conduct comprehensive experiments on these baseline methods. As shown in Table 1, we can find that, whatever the input size and the baseline model, an evident improve-

*These authors contributed equally to this work.
†Corresponding author.

Table 1. Normal angle RMSE with different neighborhood size on PCPNet dataset.

Size	DeepFit 256		DeepFit 500		DeepFit 700		AdaFit 256		AdaFit 500		AdaFit 700		GraphFit 256		GraphFit 500		GraphFit 700	
Baseline	✓		✓		✓		✓		✓		✓		✓		✓		✓	
+ Ours		✓		✓		✓		✓		✓		✓		✓		✓		✓
None	6.51	4.90	7.10	5.67	7.35	5.85	5.17	4.87	5.79	5.33	5.19	4.71	4.49	4.10	4.45	4.11	4.83	4.34
Low	9.21	8.91	9.41	9.23	9.63	9.27	9.17	9.02	9.17	9.16	9.05	8.75	8.80	8.78	8.74	8.66	8.70	8.68
Med	16.72	16.61	16.46	16.38	16.39	16.34	16.71	16.72	16.47	16.43	16.44	16.31	16.54	16.46	16.05	16.02	16.04	16.07
High	23.12	22.87	21.97	21.82	21.74	21.70	23.02	22.87	22.12	21.91	21.94	21.64	22.69	22.64	21.64	21.57	21.36	21.45
Gradient	7.31	5.52	7.71	6.22	8.06	6.43	6.03	5.72	6.64	5.84	5.90	5.51	5.15	4.91	5.22	4.83	5.51	5.31
Striped	7.92	5.70	8.66	6.51	9.26	6.92	6.00	5.79	6.30	6.01	6.01	5.48	5.28	5.00	5.48	4.89	5.61	5.35
Average	11.80	10.75	11.89	10.97	12.07	11.09	11.02	10.83	11.08	10.78	10.76	10.40	10.49	10.33	10.26	10.01	10.34	10.20

Table 2. Normal angle RMSE with different Jet order n on PCPNet dataset.

Order	DeepFit 1		DeepFit 2		DeepFit 3		GraphFit 1		GraphFit 2		GraphFit 3	
Baseline	✓		✓		✓		✓		✓		✓	
+ Ours		✓		✓		✓		✓		✓		✓
No Noise	6.72	5.25	8.08	5.20	6.51	4.90	4.70	4.22	4.62	4.17	4.45	4.11
Low Noise	9.55	9.31	9.74	9.01	9.21	8.91	8.79	8.75	8.71	8.79	8.74	8.66
Med Noise	16.77	16.77	16.56	16.61	16.72	16.61	16.29	16.30	16.11	16.02	16.05	16.02
High Noise	23.16	23.00	23.00	22.78	23.12	22.87	21.75	21.72	21.78	21.66	21.64	21.57
Gradient	7.46	5.97	8.77	5.85	7.31	5.52	5.26	5.11	5.39	4.98	5.22	4.83
Striped	7.84	6.15	9.21	6.14	7.92	5.70	5.44	5.25	5.50	5.10	5.48	4.89
Average	11.92	11.08	12.56	10.93	11.80	**10.75**	10.37	10.24	10.34	10.12	10.26	**10.01**

ment can be obtained by our methods. The results demonstrate that the proposed methods are robust and effective under different neighborhood sizes.

Robustness against the Jet orders. In addition, we provide more results on DeepFit [1] and GraphFit [3] to verify the robustness of our methods against the polynomial order n. Note that we have given the results of AdaFit [7] in the main text. Here, as shown in Table. 2, we can obtain stable performance improvements over the baseline models under different polynomial orders. Besides, we observe that $n = 3$ consistently achieves the best performance in terms of average normal angle RMSE. We consider the order 3 is suitable for most points, and our methods can reduce the approximation error of normal estimation, which bring benefits to all the baseline models under different polynomial orders.

Ablation on Z-direction Transformation Loss. As presented in Sec. 4.2 of the main text, we propose a z-direction transformation loss that constrains the transformation matrix to narrow the angle between the rotated ground-truth normal and the axis z. Here, we conduct an ablation study on z-direction transformation loss, to validate the effectiveness of our GCN-based transformation network and proposed loss function. First, we directly apply the z-direction transformation loss on original DeepFit model [1]. From the results in Table 3, we can find that the expected transformation is non-trivial for the previous transformation network, and thus there is no obvious improvements on original DeepFit model [1] (the left half of the table). Then, we replace the previous transformation network with our proposed GCN-based transformation network. In such a scenario, we can achieve evident performance improvements compared with baseline counterpart, and the best performance is obtained when the loss weight is set as 2. The experimental results imply that our z-direction transformation loss works well within a certain range.

2.2. Iterative Estimation

In the main text, we argue a better z-alignment could improve the precision of normal estimation. Here, we conduct an iterative experiment to verify it. The iterative estimation refers to feeding the normal results of n-jet fitting or error estimation module back to n-jet fitting for an estimation again. By doing so, we can rotate the estimated normal to the axis z for a better z alignment, and thus achieve a more accurate surface fitting. In Table 4, the results show a better z-alignment indeed reduces the error of normal estimation. However, simply iterative DeepFit (the 2nd line) still performs worse than our methods (the 3rd line). Besides, such scheme will cost much more inference time.

Table 3. Normal angle RMSE with different weights of z-direction transformation loss on PCPNet dataset. The left shows the results of directly applying the z-direction transformation loss on original quaternion spatial transformation network (Q-STN) [1], and the right is produced by using our GCN-based transformation network and proposed loss function.

	DeepFit						DeepFit + Ours					
Trans. Weight	0	0.1	1.0	2.0	3.0	10.0	0	0.1	1.0	2.0	3.0	10.0
No Noise	6.51	6.49	6.58	6.55	6.69	6.76	5.05	5.07	4.97	4.90	4.99	5.05
Low Noise	9.21	9.16	9.12	9.10	9.07	9.09	9.09	9.05	8.94	8.91	8.98	9.03
Med Noise	16.72	16.60	16.65	16.64	16.65	16.63	16.74	16.67	16.66	16.61	16.68	16.70
High Noise	23.12	23.02	22.96	23.03	23.02	23.05	22.88	22.87	22.86	22.87	22.90	22.90
Gradient	7.31	7.29	7.46	7.47	7.51	7.48	5.79	5.82	5.69	5.52	5.74	5.75
Striped	7.92	7.85	7.98	7.90	7.95	7.96	5.95	5.99	5.87	5.70	5.95	6.05
Average	11.80	**11.74**	11.79	11.78	11.82	11.83	10.92	10.91	10.83	**10.75**	10.87	10.91

Table 4. Iterative estimation on DeepFit model. The inference time is tested on a NVIDIA TITAN X.

Method	Average RMSE	Time (ms)
DeepFit	11.80	**0.47**
DeepFit (iterative)	11.72	0.73
DeepFit + Ours	<u>10.75</u>	<u>0.56</u>
DeepFit + Ours (iterative)	**10.72**	0.84

Table 5. L2-CD ($\times 10^4$) comparison for surface reconstruction of baseline models with or without our methods.

	DeepFit	AdaFit	GraphFit	DeepFit + Ours	AdaFit + Ours	GraphFit + Ours
Liberty	0.290	0.180	0.142	0.117	0.096	0.091
Star sharp	0.148	0.106	0.105	0.099	0.112	0.101
Column	0.212	0.195	0.212	0.218	0.193	0.179
Netsuke	0.295	0.248	0.239	0.261	0.245	0.258
Average	0.236	0.182	0.175	0.174 (**0.062**↓)	0.161 (**0.021**↓)	0.157 (**0.018**↓)

2.3. Surface Reconstruction Application

Accurate surface normals can benefit to reconstruct a better surface. So, we adopt the Poisson reconstruction implemented by Open3D library to reconstruct the surface from the point cloud with the estimated normals. Fig. 3 shows that accurate normals are helpful to reconstruct a more high-quality and complete surface from point clouds, such as the finger of liberty, and the sharp corner of star. Besides, we follow a common way [4] that samples 1×10^5 points from the reconstructed meshes and computes the L2-CD distance of them. The quantitative results are given in Table 5. The results show that our methods can help the baseline models to obtain a better reconstructed surface in most cases, and consistently achieve improvements in terms of average reconstruction error.

2.4. Visualization Results

In addition, we present more visualization results on PCPNet [2] dataset. Besides, we also provide the corresponding normal RMSE, and the percentage of good points (PGP10 and PGP5) of each shape.

References

[1] Yizhak Ben-Shabat and Stephen Gould. Deepfit: 3d surface fitting via neural network weighted least squares. In *European Conference on Computer Vision*, pages 20–34. Springer, 2020. 1, 2, 3

[2] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. Pcpnet learning local shape properties from raw point clouds. In *Computer Graphics Forum*, volume 37, pages 75–85. Wiley Online Library, 2018. 3

[3] Keqiang Li, Mingyang Zhao, Huaiyu Wu, Dong-Ming Yan, Zhen Shen, Fei-Yue Wang, and Gang Xiong. Graphfit: Learning multi-scale graph-convolutional representation for point cloud normal estimation. In *European Conference on Computer Vision*, pages 651–667. Springer, 2022. 1, 2

[4] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 2021. 3

[5] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1

[6] Xuejun Yan, Hongyu Yan, Jingjing Wang, Hang Du, Zhihong Wu, Di Xie, Shiliang Pu, and Li Lu. Fbnet: Feedback network for point cloud completion. In *European Conference on Computer Vision*, pages 676–693. Springer, 2022. 1

[7] Runsong Zhu, Yuan Liu, Zhen Dong, Yuan Wang, Tengping Jiang, Wenping Wang, and Bisheng Yang. Adafit: Rethinking learning-based normal estimation on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6118–6127, 2021. 1, 2

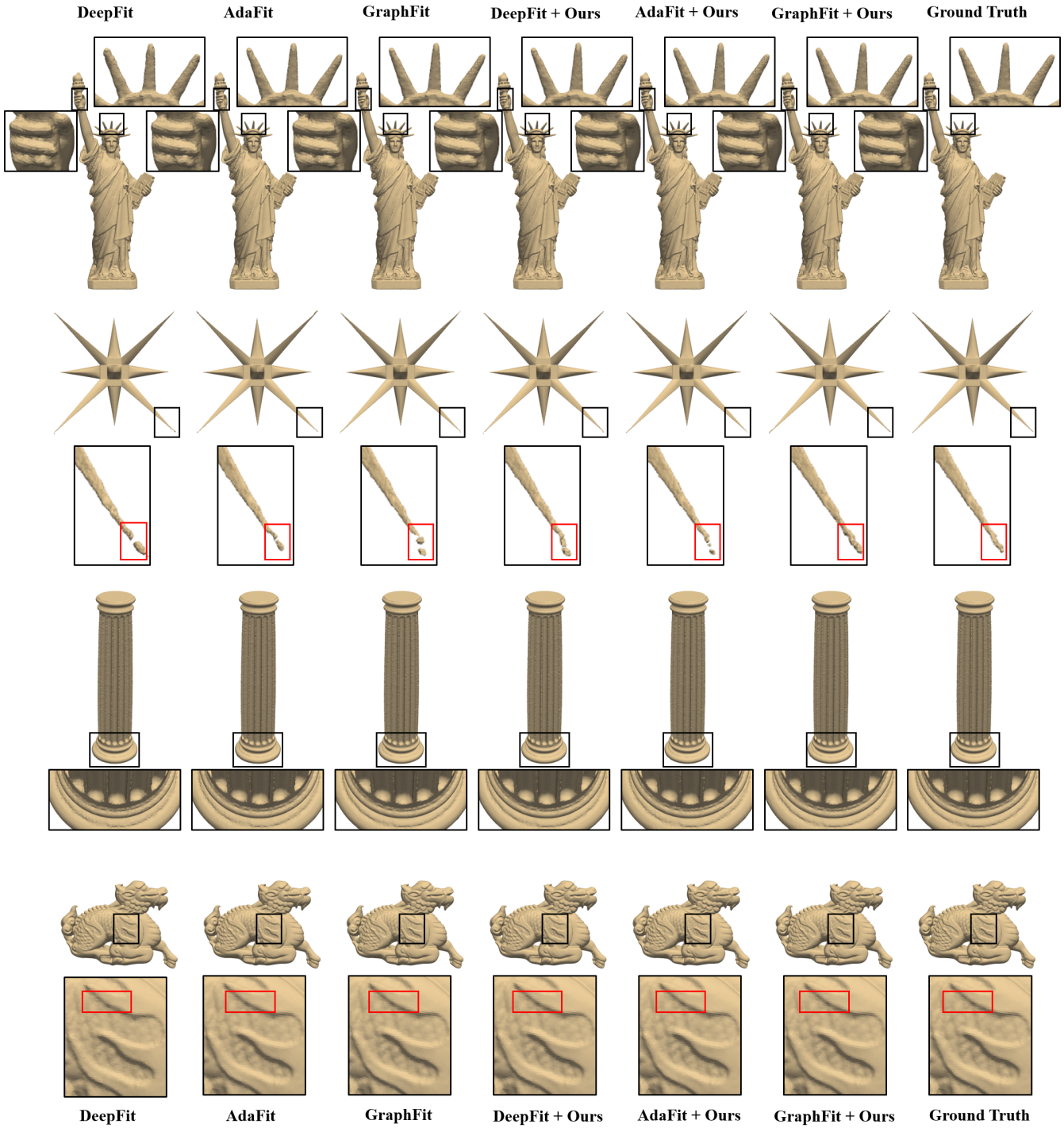

Figure 3. Comparison of surface reconstruction using the normals estimated by SOTA models with our methods. Our methods improve the baseline models to produce more accurate and complete surfaces.

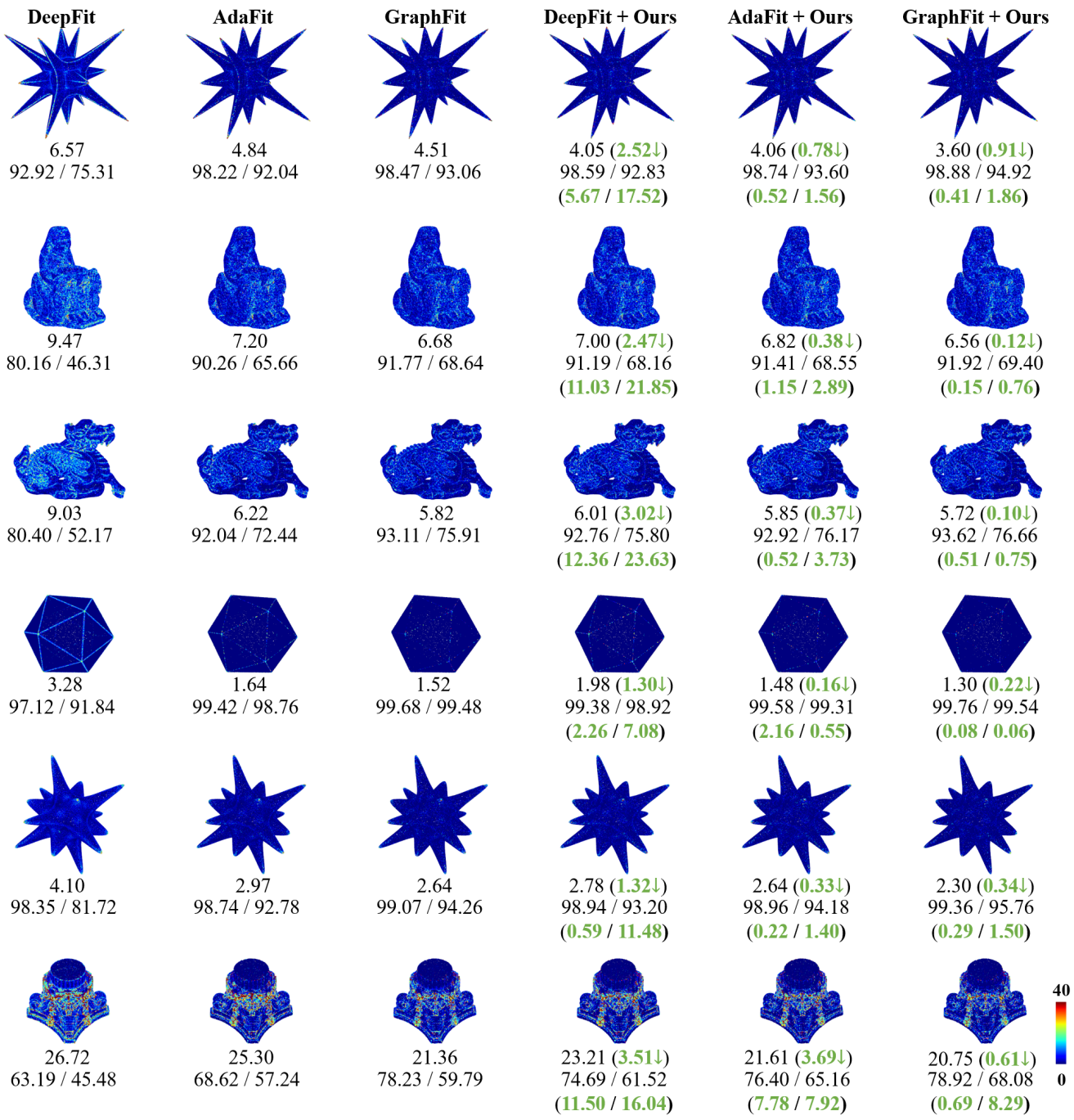

Figure 4. The error heatmap of normal estimation on PCPNet dataset. The first line of bottom values represents the average error, and second line reports the PGP10 and PGP5. The green values in brackets denote the relative improvement on the corresponding baseline models. The angle errors are mapped to a range from 0° to 40°.