Supplementary Material of Consistent Two-Flow Network for Tele-Registration of Point Clouds

Zihao Yan, Zimu Yi, Ruizhen Hu, Niloy Mitra, Daniel Cohen-Or, and Hui Huang

1 INTRODUCTION

This document provides an additional description on the data statistics (Section 2), running time and memory capacity statistics (Section 3), structure of the baseline registration network (Section 4), choices of the registration backbone, and the weights of loss function (Section 5, 6), layer details of our CTF-Net (Section 7) and more qualitative results (Section 8) for the paper submission *Consistent Two-Flow Network for Tele-Registration of Point Clouds*.

2 DATA STATISTICS

TABLE 1: Statistics of our dataset.

	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft
train	3795	1321	3256	6523	2057	2918	8183	1689
validate	100	100	100	100	100	100	100	100
test	150	150	150	150	150	150	150	150

Table 1 shows the statistics of shapes in our dataset. In total, our dataset contains 31,742 shapes in eight categories from ShapeNet v2. For each shape we process cropping 10 times to generate 10 pairs of partial shapes for training.

3 TIMING AND CAPACITY STATISTICS

TABLE 2: Comparison of the running time and model capacity of the baseline methods and our CTF-Net.

	PRNet	DCP	TopNet	MSN	PF-Net	Ours
training per shape (ms)	8.40	8.12	26.25	4.21	62.53	125.03
total training epochs	100	250	300	25	200	150
model capacity (MB)	23.79	22.54	38.13	113.08	292.22	382.46

We train CTF-Net on a computer with Intel Xeon Gold 5120 CPU at 2.20 GHz, 4 Nvidia Quadro RTX 6000, 128GB memory. The running time, training epochs, and model capacity of the baseline methods and CTF-Net are reported in Table 2. Note that the number of training epochs of each method is set according to their original settings.

Fig. 1: The architecture of the baseline registration network. Taking two paired shapes $\mathcal{S}_1$ and $\mathcal{S}_2$ and considering $\mathcal{S}_2$ as the anchor, the baseline registration network outputs parameters of a relative transformation $\mathcal{M}_{12}$ from $\mathcal{S}_1$ to $\mathcal{S}_2$.

4 STRUCTURE OF THE BASELINE REGISTRATION NETWORK

The registration network in our CTF-Net is designed to first predict the rotation parameters then the translation. To justify the superior of our registration module, we design a baseline registration network, which predicts the transformation parameters at once from the same decoder, as shown in Fig. 1.

5 CHOICES OF THE REGISTRATION BACKBONE

Here we tested our CTF-Net with a version that substitutes DCP to our registration module. The results are reported in Table 3. We observe that we achieve significantly lower translation error comparing to a version that using DCP as the registration backbone.

TABLE 3: Quantitative comparison of the CTF-Net to a version that replace our registration module with DCP network.

	CTF-Net with DCP	Ours
E_θ	15.746	15.118
E_t	13.102	5.970
E_{emd}^g	6.963	5.838
E_{emd}^f	3.124	2.399

TABLE 4: Quantitative comparisons of different combination of the loss weights.

	ω_c^{c-r}	ω_o^{c-r}	ω_c^{r-c}	ω_o^{r-c}	ω_r^{c-r}	ω_r^{r-c}	ω_{so}	ω_{sc}	ω_{sr}	ω_{st}	E_θ	E_t	E_{emd}^g	E_{emd}^f
option 1	0.5	1.5	0.5	1.5	3	3	9	3	9	9	17.403	6.954	6.912	3.038
option 2	0.5	1.5	0.5	1.5	3	3	3	1	3	3	16.699	6.735	6.859	3.012
option 3	0.5	1.5	0.5	1.5	3	3	1	0.3	1	1	17.126	7.003	6.841	2.995
option 4	0.5	1.5	0.5	1.5	3	9	3	1	3	3	15.713	6.189	6.262	2.597
option 5	0.5	1.5	0.5	1.5	3	15	3	1	3	3	16.028	6.720	6.493	2.654
option 6	1	3	0.5	1.5	3	9	3	1	3	3	15.118	5.970	5.838	2.399
option 7	1.5	4.5	1	3	3	9	3	1	3	3	15.946	6.178	5.990	2.418

6 CHOICES OF THE LOSS WEIGHTS

The full loss function with the weights can be written as:

$$L = \omega_c^{c-r} L_c^{C-R} + \omega_o^{c-r} L_o^{C-R} + \omega_c^{r-c} L_c^{R-C} + \omega_o^{r-c} L_o^{R-C} + \omega_r^{c-r} L_r^{C-R} + \omega_r^{r-c} L_r^{R-C} + \omega_{so} L_s^O + \omega_{sc} L_s^C + \omega_{sr} L_s^R + \omega_{st} L_s^T. \quad (1)$$

To further justify the setting of the loss function weights, we train CTF-Net with different weights combinations, the results are reported in Table 4. The initial weights setting option 1 is obtained base on the order of magnitude of each loss term. By comparing option 1 and 2, we observe that the performance improves as the weights of consistency losses decrease, however, if the weights of consistency losses get too low, the errors increase, as shown in option 3, thus we choose the set of consistency weights as in option 2. Similarly, we choose option 4 as the weights for the registration in R-C flow by comparing option 2, 4, 5; and choose option 6 as the weights for the completion in C-R flow by comparing option 4, 6, 7. Finally, we choose the set of weights in option 6 for Equation 1.

7 NETWORK DETAILS

The architectures and parameter numbers of our registration, completion networks are listed in Table 6 and Table 5, respectively. The structure of the orientation module inside the completion network is listed in Table 7. To avoid redundancy, we omit batch normalization layer after each convolution and linear layer. Note that our CTF-Net consists of two pairs of registration and completion networks.

To train the CTF-Net, we use the following hyper-parameters: adam optimizer, batch size = 64, learning rate = 0.001, iterations = 150.

8 MORE QUALITATIVE RESULTS

We show more qualitative results of our CTF-Net on all eight categories. Plane, car, watercraft and table categories are shown in Fig. 2; couch, cabinet, chair and lamp categories are shown in Fig. 3.

TABLE 5: Layers of the registration network. Batch size is denoted as B in the second column.

Layer (type:depth-idx)	Output Shape	Param
PointNetEncoder: 1-1	[B, 1024]	–
STN3D: 2-1	[B, 3, 3]	–
Conv1D: 3-1	[B, 64, 2048]	256
Conv1D: 3-2	[B, 128, 2048]	8,320
Conv1D: 3-3	[B, 1024, 2048]	132,096
Linear: 3-4	[B, 512]	524,800
Linear: 3-5	[B, 256]	131,328
Linear: 3-6	[B, 9]	2,313
Conv1D: 2-2	[B, 64, 2048]	256
STNkd: 2-3	[B, 64, 64]	–
Conv1D: 3-7	[B, 64, 2048]	4,160
Conv1D: 3-8	[B, 128, 2048]	8,320
Conv1D: 3-9	[B, 1024, 2048]	132,096
Linear: 3-10	[B, 512]	524,800
Linear: 3-11	[B, 256]	131,328
Linear: 3-12	[B, 4096]	1,052,672
Conv1D: 2-4	[B, 128, 2048]	8,320
Conv1D: 2-5	[B, 1024, 2048]	132,096
Linear: 1-2	[B, 512]	524,800
PointNetEncoder: 1-3	[B, 1024]	(recursive)
Linear: 1-4	[B, 512]	(recursive)
Linear: 1-5	[B, 512]	524,800
Dropout: 1-6	[B, 512]	–
Linear: 1-7	[B, 128]	65,664
Dropout: 1-8	[B, 128]	–
Linear: 1-9	[B, 4]	516
PointNetEncoder: 1-10	[B, 1024]	(recursive)
Linear: 1-11	[B, 512]	(recursive)
Linear: 1-12	[B, 512]	524,800
Dropout: 1-13	[B, 512]	–
Linear: 1-14	[B, 128]	65,664
Dropout: 1-15	[B, 128]	–
Linear: 1-16	[B, 3]	387

Fig. 2: The results of our CTF-Net on the plane, car, watercraft and table categories.

TABLE 6: Layers of the completion network. Batch size is denoted as B in the second column.

Layer (type:depth-idx)	Output Shape	Param
Latentfeature: 1-1	[B, 1920]	–
Convlayers: 2-1	[B, 1920, 1]	706,752
Convlayers: 2-2	[B, 1920, 1]	706,752
Convlayers: 2-3	[B, 1920, 1]	706,752
Conv1D: 2-4	[B, 1, 1920]	4
Linear: 1-2	[B, 1024]	3,934,208
Linear: 1-3	[B, 512]	2,098,176
Linear: 1-4	[B, 256]	524,800
Linear: 1-5	[B, 384]	196,992
Linear: 1-6	[B, 8192]	8,396,800
Conv1D: 1-7	[B, 12, 128]	780
Linear: 1-8	[B, 32768]	67,141,632
Conv1D: 1-9	[B, 64, 512]	4,160
Conv1D: 1-10	[B, 32, 512]	2,080
Conv1D: 1-11	[B, 12, 512]	396

TABLE 7: Layers of the orientation network. Batch size is denoted as B in the second column.

Layer (type:depth-idx)	Output Shape	Param
PointNetEncoder: 1-1	[B, 1024]	–
STN3D: 2-1	[B, 3, 3]	–
Conv1D: 3-1	[B, 64, 2048]	256
Conv1D: 3-2	[B, 128, 2048]	8,320
Conv1D: 3-3	[B, 1024, 2048]	132,096
Linear: 3-4	[B, 512]	524,800
Linear: 3-5	[B, 256]	131,328
Linear: 3-6	[B, 9]	2,313
Conv1D: 2-2	[B, 64, 2048]	256
Conv1D: 2-3	[B, 128, 2048]	8,320
Conv1D: 2-4	[B, 1024, 2048]	132,096
Linear: 1-2	[B, 1024]	2,099,200
Linear: 1-3	[B, 512]	2,098,176
Dropout: 1-4	[B, 512]	–
Linear: 1-5	[B, 128]	262,400
Dropout: 1-6	[B, 128]	–
Linear: 1-7	[B, 4]	1,028

Fig. 3: The results of our CTF-Net on the couch, cabinet, chair and lamp categories.