Neural Intrinsic Embedding for Non-rigid Point Cloud Matching Supplementary Materials

Puhua Jiang[1,2] Mingze Sun[1] Ruqi Huang[1]
[1]Tsinghua Shenzhen International Graduate School
[2]Peng Cheng Laboratory
{jph21, smz22}@mails.tsinghua.edu.cn ruqihuang@sz.tsinghua.edu.cn

1. Implementation Details

Basis and Descriptor Setting For a fair comparison to LIE [3], we set the dimension of the output basis of NIE to be 20. Similarly, for the matching network, NIM, we set the dimension of learned features (descriptors) to be 40. Considering NIE as a key component, we conduct an ablation study on the dimension of basis in Table 1. As dim increases, OPT is improved while GeoError is worse, which is probably due to over-fitting. Therefore setting it to be 20 is a good trade-off between the two metrics.

Down-sampling Scheme on the Modified DGCNN Recall that in Section 4 of the main submission, we propose a modified version of DGCNN [4], which leverages point cloud down-sampling for alleviating sampling density bias. We denote by n_s the size of the sub-sampled point obtained from furthest point sampling. The empirical test validates that $n_s = 3000$ achieves a good balance between efficiency and accuracy for all the datasets considered in our paper. In Fig. 1, we provide the detailed network architecture and parameters.

Dimension	10	20	30
OPT	4.7	3.1	2.9
Geo. Err	7.1	9.5	11.3

Table 1. Ablation study of basis dimension on OPT (×100), relative geodesic error (x100).

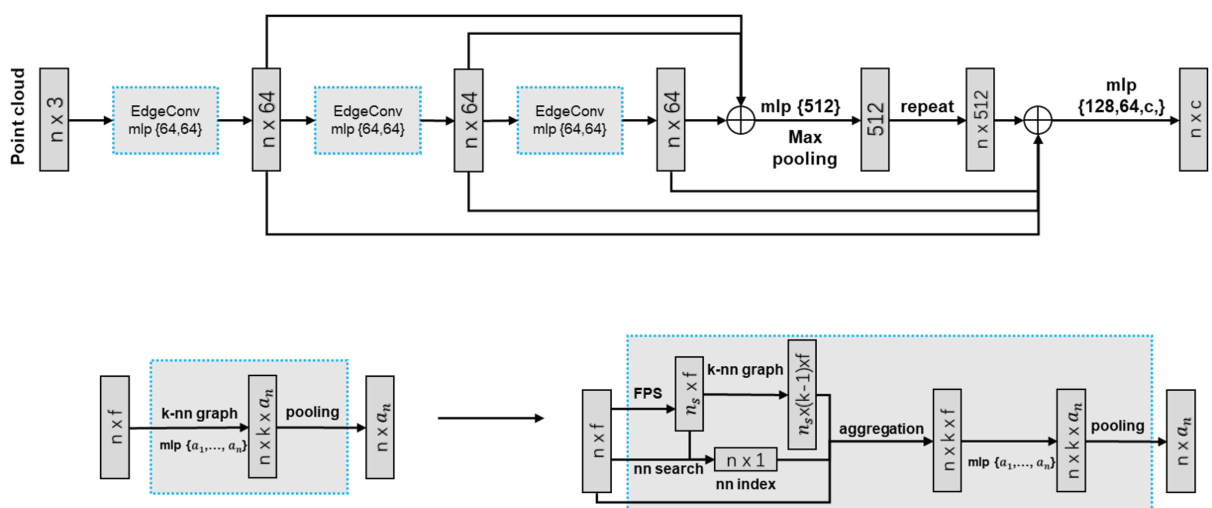

Figure 1. The top row depicts the network architecture of DGCNN [4]. In the bottom row, we specify our modification on top of the EdgeConv blocks. Taking input dimension $f = 3$ as an example: Given a point cloud X, we subsample X_s by FPS sampling. Then we conduct the nearest neighbor search of X on X_s and k-NN search(exclude itself) on X_s. After that, for a query point v_q in X, we find its nearest neighbor v_p within X_s and assign the k-NN of v_p within X_s to that of v_q. Finally, we concatenate the k-NN of v_p and v_q as the aggregation feature to the next procedure.

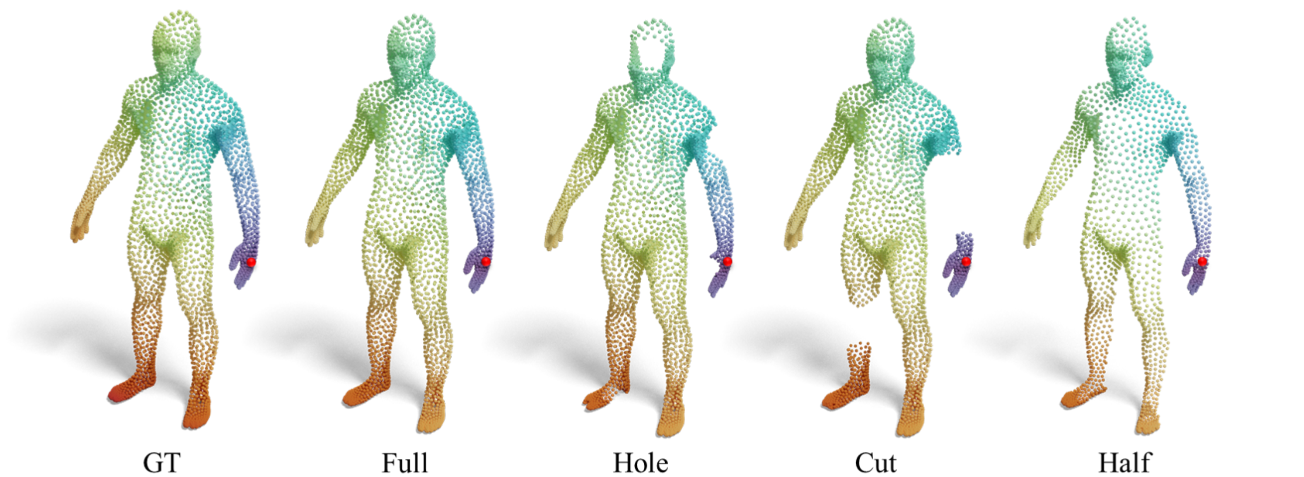

Figure 2. Geodesic distances approximated by NIE on partial point clouds. In each input, we set the source point as the red dot on the left hand, and visualize the geodesic distance from all the other points to it. The color ranges from blue (small distance) to red (large distance).

Additional details on experiments For the basis generator network, we train it with a batch size of 3 for 600 epochs. We use a cosine annealing schedule with an Adam optimizer in between a maximum learning rate of 0.002 and a minimum learning rate of 0.0002. During training, we randomly sample 4995 points from each shape. For the descriptor generator network, we use a batch size of 4, again with a cosine annealing schedule with an Adam optimizer in between a maximum learning rate of 0.002, and a minimum learning rate of 0.001. We set $\lambda_1 = 1$, $\lambda_2 = 1$ and $\lambda_3 = 0.5$ in Eqn.7 for our experiments. We run a line search of α in Eqn.11 on the small-scale dataset and fix it to be 30 for *all* experiments.

2. Geodesic Approximation for Partial Point Clouds

Unlike the prior approaches on geodesic computation [1, 2] which rely on shape connectivity, our approach can robustly approximate geodesic distances on disconnected shapes (see, e.g., the *hole* and the *cut* in Fig. 2). Table 2 shows that NIE maintains a reasonable geodesic error when partial point clouds (generated with FAUST_r dataset) are given. Fig. 2 shows the qualitative examples of geodesic distance, where the source points are all set in the left hand.

Method	full	half	hole	cut
Ours	9.5	9.8	10.	13.0

Table 2. Relative geodesic errors (×100) for full and partial point clouds.

References

[1] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Commun. ACM*, 60(11):90–99, Oct. 2017. 2

[2] R. Kimmel and J. Sethian. Fast marching methods on triangulated domains. In *PNAS*, 1998. 2

[3] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *Advances in Neural Information Processing Systems*, 33:1608–1620, 2020. 1

[4] 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