A Dataset Construction

Section 5.1 has explained how to generate the source point cloud and the target point cloud on Scan2CAD [2]. Here, we mainly focus on how to generate the putative correspondences by feature matching. We used the FCGF [13] as feature extractor, whose parameter setting is shown in Table 5. The FCGF network is pretrained on 3DMatch dataset and then fine-tuned with parameter setting in Table 6. The fine-tuned FCGF extracts L2-normalized local feature $F_X^{local} = \left\{f_{x_i}^{local} \in R^{32}\right\}_{i=1}^{|X|}$ for the source point cloud X and $F_Y^{local} = \left\{f_{y_i}^{local} \in R^{32}\right\}_{i=1}^{|Y|}$ for the target point cloud Y, where $|X|$ and $|Y|$ denote the number of points in the source point cloud and the target point cloud, respectively. Given each point $y_j \in Y$ in the target point cloud, we find the point $x_i \in X$ in the source point cloud satisfying $i = \arg\max_i < f_{y_j}^{local}, f_{x_i}^{local} >$ to build correspondence (x_i, y_j), where $< f_{y_j}^{local}, f_{x_i}^{local} >$ is the cosine similarity between two point features. In this way, we obtain $|Y|$ correspndences and we define the cosine similarity $< f_{x_i}^{local}, f_{y_j}^{local} >$ as the saliency score of correspondence (x_i, y_j). We select K correspondences with largest saliency scores and then randomly downsample these K correspondences to N correspondences as the input putative correspondences. Here, we set K as 10000 to make input correspondences cover as many instances as possible. N is set to 1000, which has been stated in section 5.1.

Table 5. Parameter setting and pretraining of the FCGF network.

Model	RESUNETBN2C
Downsampling voxel size	2.5cm (0.025)
Feature dimension	32
Pretrained dataset	3DMatch
Normalized feature	True

Table 6. Parameter setting in fine-tuning the FCGF network.

Batch size	4
Learning rate	10^{-3}
Epoch	20
Optimizer	SGD

The per-instance inlier ratio of the input putative correspondences for the Scan2CAD dataset is show in Figure 5. We can see that most instances have an inlier rate of less than 10%, and many are less than 2%.

B Hyperparameters Choice

The threshold τ_S and τ_N are key hyperparameters of our proposed pruning strategy, which are used to binarize the compatibility between correspondences and

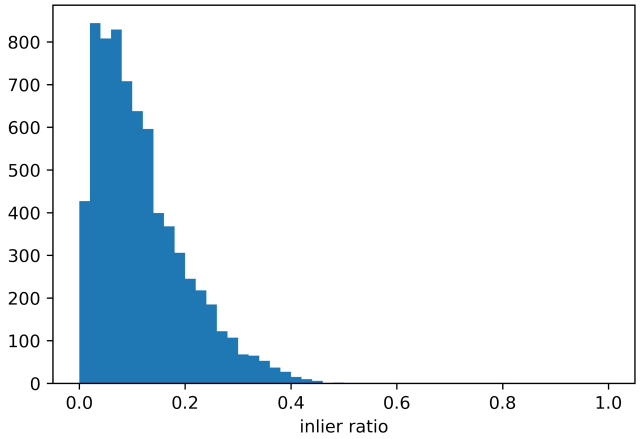


Fig. 5. The histogram of per-instance inlier ratio on Scan2CAD.

identity the inlier sets. We first evaluate the performance of our framework with different thresholds τ_S and the results are show in Table 7. Our framework is robust to the choices of threshold τ_S, and the performance of our framework is significantly superior to existing methods with all different values of it. This is because our correspondences are well separable in the feature space. We calculate the average cosine similarity between correspondences in the feature space and the results are shown in Table 8. In Table 8, Positive denotes the average cosine similarity between anchor correspondences and their positive samples in the feature space. Top-K denotes the average cosine similarity between anchor correspondences and their top-K hardest samples in the feature space. There exists a large margin between Positive and Top-K, which indicates the correspondences are well separated in the feature space. The choice of threshold τ_S is a tradeoff, which slightly affected the performance of our framework. If we choose a smaller τ_S, fewer correspondences will be pruned, which may improve the recall but decrease the precision.

Table 7. Performance of our PointCLM when varying the threshold τ_S.

	MR(%)	MP(%)	MF(%)
0.70	78.54	68.29	73.06
0.75	**79.35**	68.81	73.71
0.80	78.81	68.64	73.37
0.85	78.10	**70.64**	**74.18**
0.90	74.06	**70.64**	72.31

Table 8. Average cosine similarity within positive pairs and top-K hardest negative pairs in the feature space.

Positive	Top-1 (%)	Top-5 (%)	Top-10 (%)	Top-15 (%)
83.96	61.32	52.90	49.65	47.59

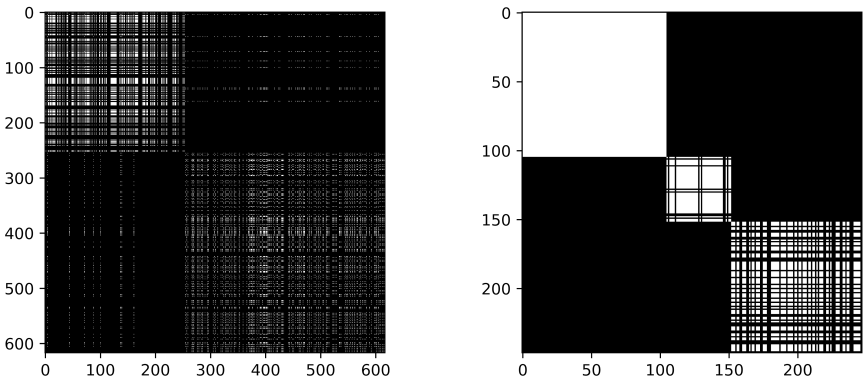
Then we evaluate the performance of our framework with different τ_N and the results are shown in Table 9. Again, our framework outperforms all existing methods with all different choices of τ_N, though it has some influence on each metric. The choice of the threshold τ_N is also a trade-off. If we choose a smaller τ_N, some outlier clusters may be considered as instances and some instances with small inlier ratios may be registered successfully, which results in a lower precision and a higher recall. In practice, fine-tuning of the parameters can be performed on a validation set.

Table 9. Performance of our PointCLM when varying the threshold τ_N.

	MR(%)	MP(%)	MF(%)
10	**80.60**	61.44	69.73
15	78.74	66.82	72.29
20	78.10	70.64	**74.18**
30	76.56	**71.28**	73.83

C Visualization

We quantitatively analyze the role of deep representations in Section 5.4. Here, we visualize the clustering results with and without deep representation in Figure 6 to demonstrate the effect of our deep representation qualitatively. We select an example, whose target point cloud contains three instances. We first use a 3-dimensional one-hot vector to represent which instance a correspondence belongs to. Then we use these vectors to calculate similarity matrices and permute these matrices with the results of the clustering. It can be seen that the similarity matrix permuted by the framework with deep representation is much smaller than the one permuted by the framework without deep representation because more outliers are removed during pruning in the former case. More importantly, the matrix in Figure 6(b) shows three clear clusters, which correspond to the three instances. On the contrary, the matrix in Figure 6(a) shows two blocks, where the lower right block actually corresponds to two instances, which cannot be distinguished successfully without using the proposed deep representation.



(a) Without deep representation (b) With deep representation

Fig. 6. Visualization of clustering results without and with deep representation.