

DeepGMR: Learning Latent Gaussian Mixture Models for Registration –Supplementary Material–

Wentao Yuan¹ Benjamin Eckart² Kihwan Kim²
Varun Jampani² Dieter Fox^{1,2} Jan Kautz²

¹ University of Washington

² NVIDIA

1 Overview

In this document, we provide additional details, discussions, and experiments to support the original submission. Below is a summary of the contents.

- Sec. 2 provides detailed derivation and proofs for the \mathbf{M}_T compute block described in Sec. 4.3 of the main paper.
- Sec. 3 contains auxiliary results, including error distribution curves (Fig. 1), ablation studies (Fig. 2), robustness tests (Fig. 3) and category-specific results (Table 1) in support of the major results in Sec. 5 of the main paper.
- Sec. 4 shows additional visualizations of registration results (Fig. 5) and the learned latent GMMs (Fig. 6).
- Sec. 5 discusses a limitation of our method and suggests directions for future research.

2 Additional Derivation

2.1 KL-divergence to Maximum Likelihood

We prove the conditions whereby Eq. (12) in the main paper is equivalent to Eq. (13), *i.e.* the conditions under which minimizing the KL-divergence from the transformed source distribution $T(\hat{\Theta})$ to the target distribution Θ is equivalent to maximizing the likelihood of the transformed source point cloud $T(\hat{P})$ under the target distribution Θ .

Given two probability distributions $p(x)$ and $q(x)$ on \mathcal{X} , the KL-divergence between p and q is defined to be

$$\text{KL}(p \mid q) = \int_{\mathcal{X}} p(x) [\ln p(x) - \ln q(x)] dx \quad (1)$$

$$= \mathbb{E}_{x \sim p(x)} \ln p(x) - \mathbb{E}_{x \sim p(x)} \ln q(x) \quad (2)$$

Thus, we can write the KL-minimization problem in Eq. (12) of the main paper as follows,

$$T^* = \operatorname{argmin}_T \operatorname{KL}(T(\hat{\Theta}) | \Theta) \quad (3)$$

$$= \operatorname{argmin}_T \mathbb{E}_{x \sim p(x|T(\hat{\Theta}))} \ln p(x | T(\hat{\Theta})) - \mathbb{E}_{x \sim p(x|T(\hat{\Theta}))} \ln p(x | \Theta) \quad (4)$$

Note that the first term, the negative entropy of $p(x|T(\hat{\Theta}))$, is invariant with respect to T , so we end up with

$$T^* = \operatorname{argmax}_T \mathbb{E}_{x \sim p(x|T(\hat{\Theta}))} \ln p(x | \Theta) \quad (5)$$

Thus, minimizing the KL-divergence from $T(\hat{\Theta})$ to Θ is equivalent to maximizing the expected log likelihood of data distributed according to $T(\hat{\Theta})$ under Θ . Or, in other words, minimizing the cross-entropy. Note that the transformed source point cloud $T(\hat{\mathcal{P}}) = \{T(\hat{p}_i)\}_{i=1}^N$ are sampled *iid* from the distribution $p(x|T(\hat{\Theta}))$. Using the law of large numbers, given a suitably large point cloud, we can approximate the expectation in Eq. (5) as

$$\operatorname{argmax}_T \mathbb{E}_{x \sim p(x|T(\hat{\Theta}))} \ln p(x|\Theta) \approx \operatorname{argmax}_T \frac{1}{N} \sum_{i=1}^N \ln p(T(\hat{p}_i) | \Theta) \quad (6)$$

$$= \operatorname{argmax}_T \sum_{i=1}^N \ln \sum_{j=1}^J \pi_j \mathcal{N}(T(\hat{p}_i) | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \quad (7)$$

which gives us Eq. (13) in the main paper.

2.2 Single Sum Reduction

We show how to reduce the NJ pairs of distances in Eq. (16) of the main paper to J pairs of distances in Eq. (17) of the main paper using the output of \mathbf{M}_Θ (Eqs. (7,8,9) in the main paper).

The calculations inside \mathbf{M}_Θ (Eqs. (7,8,9) in the main paper) determine the relationships between the correspondence matrix $\hat{I} = \{\hat{\gamma}_{ij}\}_{i,j=1,1}^{N,J}$, the point coordinates $\hat{\mathcal{P}} = \{\hat{p}_i\}_{i=1}^N$ and the GMM parameters $\hat{\Theta} = \{\hat{\pi}_j, \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j\}_{j=1}^J$. Specifically, we can rewrite Eqs. (7,8) in the main paper as

$$\sum_{i=1}^N \hat{\gamma}_{ij} = N \hat{\pi}_j \quad (8)$$

$$\sum_{i=1}^N \hat{\gamma}_{ij} T(\hat{p}_i) = N \hat{\pi}_j T(\hat{\boldsymbol{\mu}}_j) \quad (9)$$

To prove the latter identity, note that the 3D rigid transformation T is a linear operator. Therefore,

$$\sum_{i=1}^N \hat{\gamma}_{ij} T(\hat{p}_i) = T \left(\sum_{i=1}^N \hat{\gamma}_{ij} \hat{p}_i \right) \quad (10)$$

$$= T(N\hat{\pi}_j \hat{\boldsymbol{\mu}}_j) \quad (11)$$

$$= N\hat{\pi}_j T(\hat{\boldsymbol{\mu}}_j) \quad (12)$$

Next, we expand the right hand side of Eq. (16) in the main paper, which contains NJ pairs of distances, using Eqs. (8,9).

$$\sum_{i=1}^N \sum_{j=1}^J \hat{\gamma}_{ij} \|T(\hat{p}_i) - \boldsymbol{\mu}_j\|_{\boldsymbol{\Sigma}_j}^2 \quad (13)$$

$$= \sum_{j=1}^J \sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{p}_i)\|_{\boldsymbol{\Sigma}_j}^2 - 2 \sum_{j=1}^J \boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}_j^{-1} \sum_{i=1}^N \hat{\gamma}_{ij} T(\hat{p}_i) + \sum_{j=1}^J \|\boldsymbol{\mu}_j^\top\|_{\boldsymbol{\Sigma}_j}^2 \sum_{i=1}^N \hat{\gamma}_{ij} \quad (14)$$

$$= \sum_{j=1}^J \left(\sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{p}_i)\|_{\boldsymbol{\Sigma}_j}^2 - 2\boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}_j^{-1} N\hat{\pi}_j T(\hat{\boldsymbol{\mu}}_j) + \|\boldsymbol{\mu}_j^\top\|_{\boldsymbol{\Sigma}_j}^2 N\hat{\pi}_j \right) \quad (15)$$

Now, we complete the square by adding $N\hat{\pi}_j \|T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2$ to the latter two terms in the summation and subtracting it from the first term. For the latter two terms, we have

$$N\hat{\pi}_j \|T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 - 2\boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}_j^{-1} N\hat{\pi}_j T(\hat{\boldsymbol{\mu}}_j) + \|\boldsymbol{\mu}_j^\top\|_{\boldsymbol{\Sigma}_j}^2 N\hat{\pi}_j \quad (16)$$

$$= N\hat{\pi}_j \left(\|T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 - 2\boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}_j^{-1} T(\hat{\boldsymbol{\mu}}_j) + \|\boldsymbol{\mu}_j^\top\|_{\boldsymbol{\Sigma}_j}^2 \right) \quad (17)$$

$$= N\hat{\pi}_j \|\boldsymbol{\mu}_j - T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 \quad (18)$$

For the first term, we have

$$\sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{p}_i)\|_{\boldsymbol{\Sigma}_j}^2 - N\hat{\pi}_j \|T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 \quad (19)$$

$$= \sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{p}_i)\|_{\boldsymbol{\Sigma}_j}^2 - 2N\hat{\pi}_j \|T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 + N\hat{\pi}_j \|T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 \quad (20)$$

$$= \sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{p}_i)\|_{\boldsymbol{\Sigma}_j}^2 - 2N\hat{\pi}_j T(\hat{\boldsymbol{\mu}}_j)^\top \boldsymbol{\Sigma}_j^{-1} T(\hat{\boldsymbol{\mu}}_j) + \sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 \quad (21)$$

$$= \sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{p}_i)\|_{\boldsymbol{\Sigma}_j}^2 - \sum_{i=1}^N \hat{\gamma}_{ij} T(\hat{p}_i)^\top \boldsymbol{\Sigma}_j^{-1} T(\hat{\boldsymbol{\mu}}_j) + \sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 \quad (22)$$

$$= \sum_{i=1}^N \hat{\gamma}_{ij} \|T(\hat{p}_i) - T(\hat{\boldsymbol{\mu}}_j)\|_{\boldsymbol{\Sigma}_j}^2 \quad (23)$$

Eq. (21) and (22) uses the relationship in Eq. (8) and (9) respectively. Notice that the result in Eq. (23) is invariant to T because we assume Σ_j is isotropic. Therefore, if we are optimizing over T , we can reduce Eq. (15), *i.e.* the right hand side of Eq. (16) in the main paper, to the single term in Eq. (18), which gives us

$$T^* = \operatorname{argmin}_T \sum_{i=1}^N \sum_{j=1}^J \hat{\gamma}_{ij} \|T(\hat{\rho}_i) - \boldsymbol{\mu}_j\|_{\Sigma_j}^2 \quad (24)$$

$$= \operatorname{argmin}_T \sum_{j=1}^J N \hat{\pi}_j \|\boldsymbol{\mu}_j - T(\hat{\boldsymbol{\mu}}_j)\|_{\Sigma_j}^2 \quad (25)$$

$$= \operatorname{argmin}_T \sum_{j=1}^J \frac{\hat{\pi}_j}{\sigma_j^2} \|T(\hat{\boldsymbol{\mu}}_j) - \boldsymbol{\mu}_j\|^2 \quad (26)$$

This is exactly Eq. (17) in the main paper.

2.3 SVD Solution

We derive the solution to the weighted ICP criterion in Eq. (17) of the main paper using a weighted version of Umeyama's method [5]. First, we center the data and construct the cross-covariance matrix M ,

$$\boldsymbol{\mu}_c \stackrel{\text{def}}{=} \sum_{j=1}^J \hat{\pi}_j \boldsymbol{\mu}_j \quad (27)$$

$$\hat{\boldsymbol{\mu}}_c \stackrel{\text{def}}{=} \sum_{j=1}^J \hat{\pi}_j \hat{\boldsymbol{\mu}}_j \quad (28)$$

$$M \stackrel{\text{def}}{=} \sum_{j=1}^J \frac{\hat{\pi}_j}{\sigma_j^2} (\boldsymbol{\mu}_j - \boldsymbol{\mu}_c)(\hat{\boldsymbol{\mu}}_j - \hat{\boldsymbol{\mu}}_c)^T \quad (29)$$

Assuming $T \in SE(3)$, given the SVD decomposition of $M = USV^T$, the optimal rotation R^* and translation \mathbf{t}^* are as follows,

$$R^* = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det VU^T \end{bmatrix} U^T \quad (30)$$

$$\mathbf{t}^* = \boldsymbol{\mu}_c - R^* \hat{\boldsymbol{\mu}}_c \quad (31)$$

The center matrix in Equation (30) comes from the fact that we want to enforce $\det R^* = +1$ to prevent reflections.

3 Additional Quantitative Results

3.1 Full Error Distribution

Fig. 1 contains the cumulative distribution function (CDF) curves of the RMSE metric for the methods tested in Sec. 5 of the main paper. To be specific, a point (x, y) on the curve implies that fraction y of the instances in the test set has RMSE less than x . The CDF curves show the complete error distribution which reveals more information than a single metric. In fact, recall@0.2 shown in the main paper is a single point on the CDF curve with $x = 0.2$.

A couple of observations can be drawn from the error distribution

- Some local methods such as ICP [3] and HGMR [4] are quite accurate on a fraction of instances (in particular, those with small transformations).
- Methods based on point-to-point (ICP [3], DCP [6], PRNet [7]) and feature correspondences (FGR [8], PointNetLK [1]) performs worse on noisy data, whereas methods based on probabilistic data association (HGMR [4], DeepGMR) are unaffected.
- Learning-based methods (PointNetLK [1], DCP [6], PRNet [7]) except DeepGMR perform significantly worse on data from unseen categories, which shows that the generalization to unseen data demonstrated in these works does not hold in the case of unrestricted rotation.

3.2 Ablation Studies

We perform ablations on several design choices mentioned in Sec. 4 of the main paper, including the number of GMM components J , the input to correspondence network f_ψ and the loss function. The dataset used is ModelNet noisy. The results are compared using CDF of RMSE on the test set shown in Fig. 2.

Number of GMM components It can be seen that the performance of DeepGMR saturates with $J > 16$, so we use $J = 16$ across our experiments.

Input feature While the performance of DeepGMR is indeed improved with RRI features [2], DeepGMR taking raw xyz coordinates still outperforms the most competitive baseline DCP.

Loss function We trained DeepGMR directly with the RMSE metric used for evaluation (Eq. 20 of the main paper) and compared it to DeepGMR trained with the simple MSE loss in Eq. 18 of the main paper. We found the two models perform almost identically (the MSE-trained model has slightly lower maximum error). This shows that DeepGMR is not sensitive to the particular choice of loss function. Although it is possible to design a better loss function, it is not the focus of our work.

3.3 Robustness Tests

We perform additional tests on the robustness of DeepGMR to input point density and transformation magnitude. The results are shown in Fig. 3.

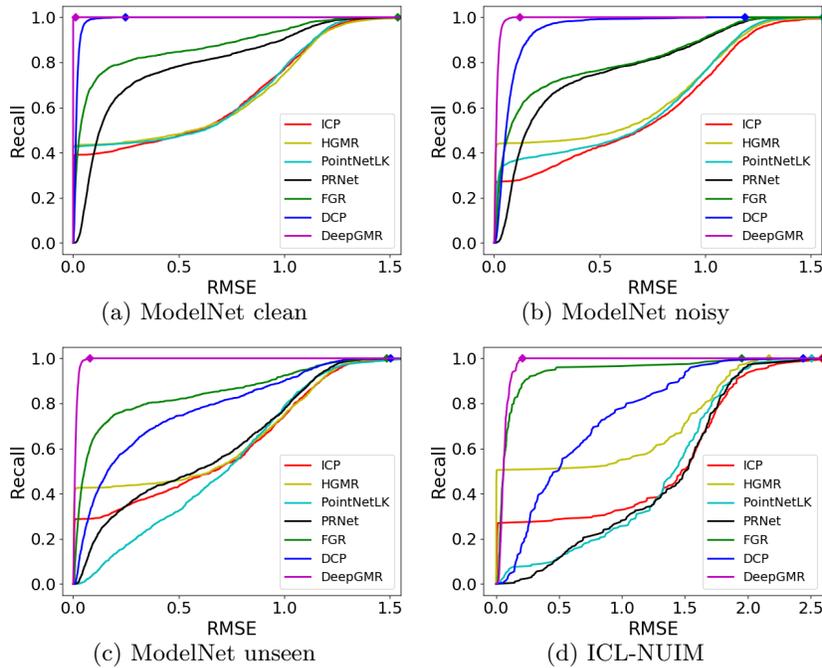


Fig. 1: Cumulative distribution function (CDF) of RMSE metric on the test set of the evaluation datasets in Sec. 5 of the main paper. A point (x, y) on the curve indicates the method achieve a recall of y with threshold x on that dataset. The diamonds show where the CDF reaches 1, i.e. the maximum error across the entire test set for that method. If there is no diamond, it means the maximum error is beyond the x -axis limit.

Point density Because DeepGMR performs registration in the latent GMM space, it is invariant to the density of input point clouds. To demonstrate this, we test the DeepGMR model trained on ModelNet noisy in Sec. 5 of the main paper on point clouds with various density without any finetuning. Here, we can use the number of points N as a proxy for density since the point clouds are uniformly sampled from same surface. From the results in Fig. 3a, it can be seen that the performance of DeepGMR is unaffected on point clouds up to 4 times denser than training and is only slightly worse on point clouds up to 4 times sparser, which can be attributed to missing geometric details in sparse point clouds.

We note that the accuracy of methods that depend on hand-crafted feature correspondences, e.g. FGR, may improve with more input points as better normals can be estimated. However, we found that this is only true on data without noise. With 4096 input points, the accuracy of FGR improves on ModelNet clean (0.04 RMSE, 0.96 recall), but stays the same on ModelNet noisy (0.22 RMSE, 0.77 recall). This test indicates that FGR’s normal estimation accuracy is more contingent on the noise level than on the sampling density.

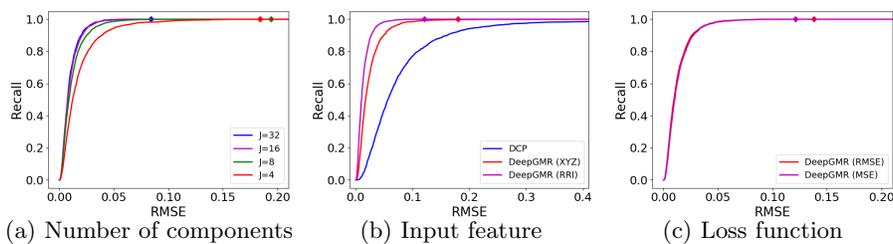


Fig. 2: Ablation studies on **(a)** the number of GMM components J , **(b)** the input to correspondence network (RRI [2] or raw XYZ) and **(c)** loss function. Ablated models are compared using CDF of RMSE on the ModelNet noisy test set. The diamonds show where the CDF reaches 1, i.e. the maximum error across the entire test set for that method.

Input transformation magnitude DeepGMR learns latent correspondences between points and GMM components that are *pose-invariant*, which means that its output is invariant to the magnitude of the transformation between the input point clouds. From Fig. 3b, we can see that DCP [6], another learning-based global method, shares the same invariance property while the performance of local methods (ICP [3], HGMR [4], PointNetLK [1]) degrades significantly with larger transformation. We also observe an interesting class of registration methods including FGR [8] and PRNet [7]. The formulations of these methods are global but they rely on feature matching or keypoint detection that become unstable with larger transformation, which makes their performance worse on these cases.

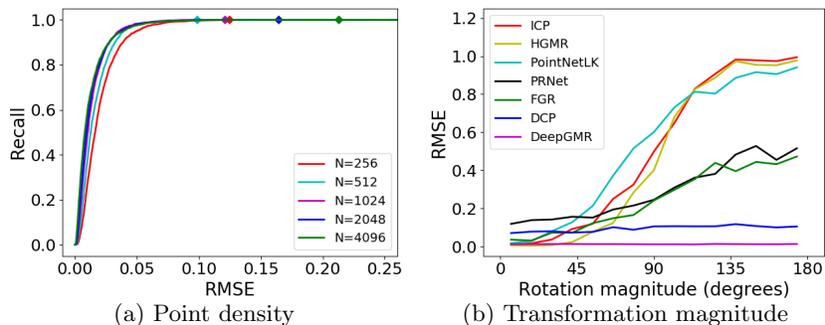


Fig. 3: Robustness tests. **(a)** RMSE CDF curves of the same DeepGMR model (trained with $N = 1024$) tested on point clouds of different density (density is measured by the number of subsampled points N). **(b)** RMSE vs magnitude of ground truth rotation between source and target on ModelNet noisy test set.

3.4 Per-category Results

Table 1 compares the average RMSE within each of the 40 categories in ModelNet noisy. DeepGMR achieves consistently good performance across all categories, while the baseline methods struggle with objects that have rotation or reflection symmetry (e.g. bowl, glass box), repetitive structure (e.g. bookshelf, stairs) or thin parts (e.g. radio, lamp).

4 Additional Qualitative Results

4.1 Registration Results

More qualitative registration results on ModelNet noisy and ICL-NUIM are shown in Fig. 5. It can be seen that DeepGMR is able to deal with challenging cases that trap other methods in local minima, such as repetitive structure, undersampled thin parts and non-planar geometry, which demonstrate the consistency and robustness of the correspondences learned by DeepGMR.

4.2 GMM Visualization

We show more visualization of the learned GMM and correspondence in Fig. 6. We can see that different object parts are assigned to different GMM components consistently across views. Note that no explicit supervision is provided on the correspondence. Everything is learned end-to-end with the registration objective.

5 Future Work

One limitation of DeepGMR is that it does not explicitly consider partial overlap, i.e. when the IoU between source and target point clouds is less than 1 after alignment. The reason is that DeepGMR estimates the correspondence between *all* points and *all* components in the latent GMM. In the case of partial overlap, however, it is more ideal to estimate a partial correspondence, i.e. the correspondence between *some* of the points and *some* of the components in the latent GMM.

To measure the consequence of this limitation, we performed a preliminary experiment on partial data artificially created from ModelNet40. Specifically, we generate partial point clouds by approximating the rendering procedure of an orthographic depth camera. First, we randomly rotate the complete point cloud and project the points onto a zero-centered grid of dimension 200×200 and size 2×2 (same size as the bounding box of the point cloud, which is normalized to $[-1, 1]^3$ across the dataset) on the xy -plane. Then, for each grid cell, we keep one point with the smallest z value and throw away the others. In this way, we end up with a partial point cloud that closely resembles the observation of a depth camera. Finally, we add independent Gaussian noise to the points.

Experimental results on this partial dataset are summarized in Fig. 4. On one hand, we note that even though DeepGMR does not explicitly consider partial overlap, its performance is still competitive. In addition, if we apply a refinement stage afterwards (i.e. use the prediction of a global method as the initialization of a local method such as ICP), DeepGMR achieves the best performance on this partial dataset. This demonstrates the power of the robust data association learned by DeepGMR. On the other hand, PRNet [7], a prior work that explicitly considers partial overlap, fails on this dataset. This shows that their method of dealing with partial overlap only works with limited transformation magnitude, i.e. it is a local registration method.

Although DeepGMR is able to outperform baselines on partial overlap data with the help of local refinement, its performance is still far below its performance on completely overlapping data. Therefore, a promising future research direction is to combine the robust point-to-latent-GMM correspondence learned by DeepGMR with techniques that deal with partial overlap (e.g. the attention mechanisms in [6,7]).

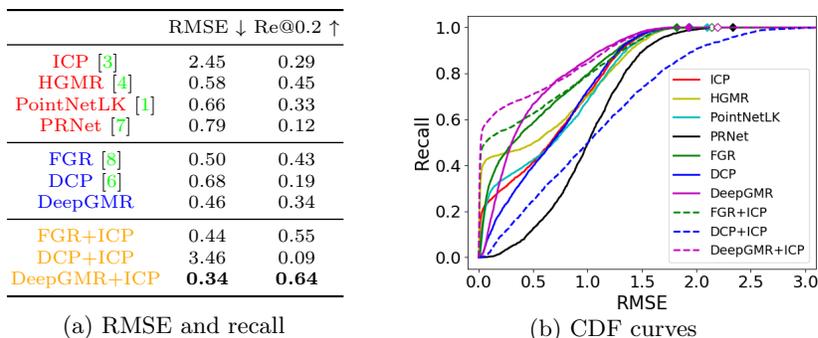


Fig. 4: Results on ModelNet partial: **(a)** Average RMSE and recall with threshold 0.2; **(b)** CDF of RMSE. **Local** methods outperform **global** methods on a fraction of instances with small transformations but fail on the remaining ones. DeepGMR+ICP, a **global+local** method that uses the output of DeepGMR as the initialization for ICP, achieves the best overall performance. Although DeepGMR by itself is not as accurate as in the case of complete overlap, it is able to bring most instances in the convergence basin of local methods. Best viewed in color.

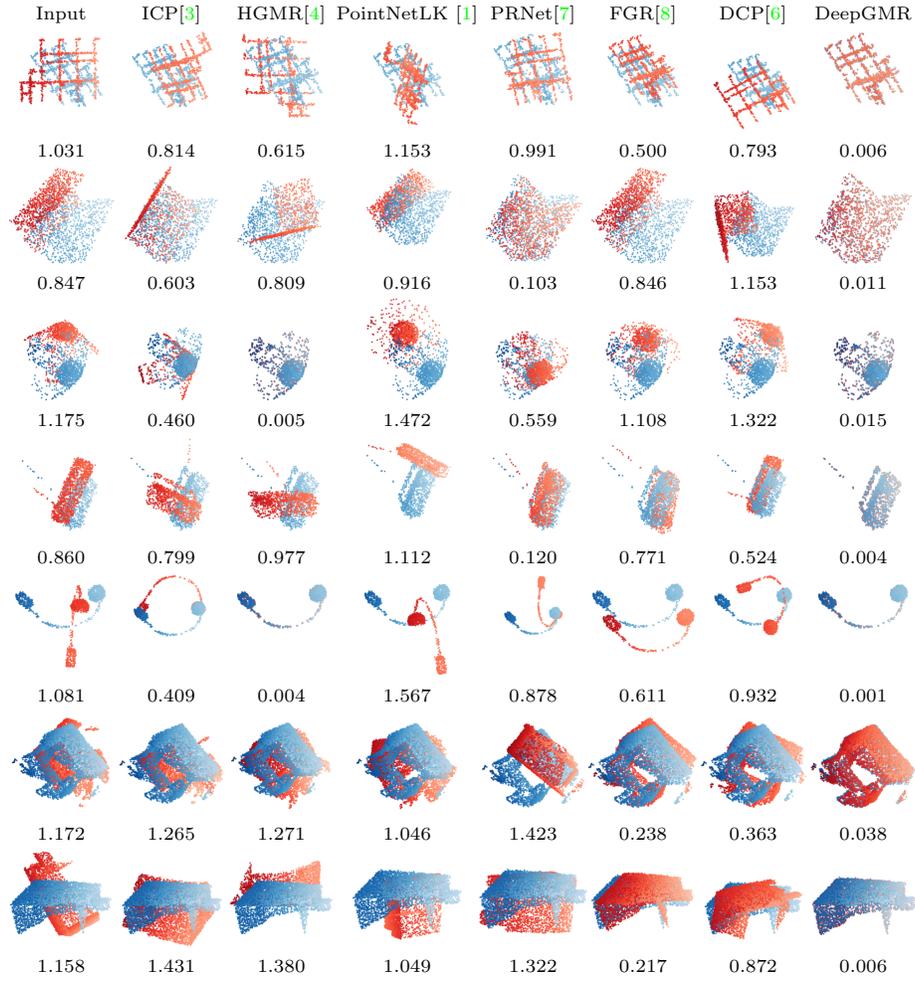


Fig. 5: Qualitative registration results on ModelNet40 noisy (top 5 rows) and ICL-NUIM point clouds (bottom 2 rows). The RMSE of each example is labeled below the plot. These examples highlight some typical failure modes of existing methods such as 1) ignoring parts with sparse point samples 2) erroneous data association due to repetitive structures and symmetry. DeepGMR avoids these errors by estimating consistent point-to-distribution correspondence and performing robust registration on GMMs.

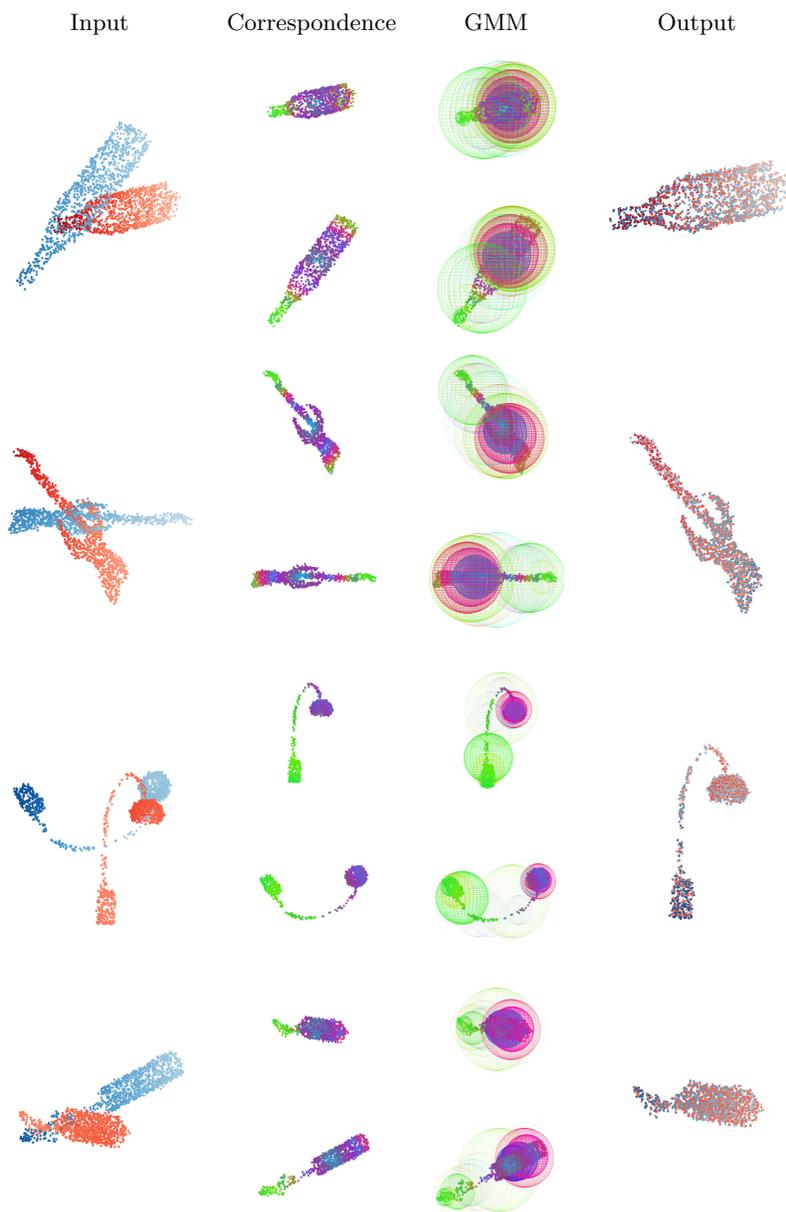


Fig. 6: Visualization of learned correspondences and GMMs. In the second and third column, each color indicates a different GMM component. The point colors are calculated as weighted averages of the component colors according to the learned correspondences. The radius of each sphere is equal to the standard deviation of the GMM component. Note how DeepGMR learns to correspond the points and GMMs in the source (red in first column, top in each row) and target (blue in first column, bottom in each row) without any explicit supervision.

Table 1: Comparison of average RMSE within each category on ModelNet noisy

Category	ICP[3]	HGMR[4]	PointNetLK [1]	PRNet[7]	FGR[8]	DCP[6]	DeepGMR
airplane	0.49	0.48	0.51	0.24	0.15	0.08	0.01
bathtub	0.53	0.43	0.62	0.31	0.30	0.13	0.01
bed	0.46	0.66	0.52	0.20	0.24	0.08	0.01
bench	0.51	0.62	0.42	0.41	0.38	0.15	0.02
bookshelf	0.58	0.50	0.56	0.33	0.22	0.14	0.01
bottle	0.51	0.55	0.51	0.27	0.25	0.07	0.01
bowl	0.77	0.76	0.80	0.57	0.61	0.15	0.02
car	0.41	0.44	0.47	0.23	0.16	0.09	0.01
chair	0.55	0.51	0.55	0.25	0.20	0.09	0.01
cone	0.50	0.77	0.54	0.24	0.30	0.13	0.02
cup	0.64	0.67	0.72	0.48	0.39	0.10	0.01
curtain	0.58	0.41	0.41	0.34	0.36	0.10	0.01
desk	0.58	0.54	0.53	0.21	0.18	0.11	0.01
door	0.55	0.53	0.55	0.26	0.60	0.14	0.02
dresser	0.57	0.51	0.59	0.34	0.23	0.10	0.02
flower pot	0.27	0.51	0.37	0.25	0.15	0.09	0.01
glass box	0.65	0.68	0.61	0.41	0.47	0.11	0.02
guitar	0.40	0.52	0.36	0.31	0.47	0.05	0.00
keyboard	0.55	0.44	0.53	0.47	0.45	0.07	0.01
lamp	0.59	0.78	0.37	0.27	0.18	0.07	0.01
laptop	0.32	0.53	0.51	0.37	0.37	0.09	0.01
mantel	0.56	0.48	0.56	0.27	0.23	0.08	0.01
monitor	0.61	0.48	0.59	0.27	0.30	0.10	0.01
night stand	0.63	0.58	0.56	0.36	0.27	0.11	0.01
person	0.43	0.27	0.42	0.27	0.19	0.07	0.00
piano	0.55	0.58	0.59	0.24	0.15	0.08	0.01
plant	0.51	0.50	0.47	0.27	0.21	0.07	0.01
radio	0.37	0.71	0.43	0.26	0.33	0.09	0.01
range hood	0.56	0.50	0.54	0.31	0.25	0.07	0.01
sink	0.51	0.60	0.54	0.30	0.16	0.09	0.01
sofa	0.43	0.50	0.49	0.35	0.18	0.10	0.01
stairs	0.63	0.53	0.58	0.29	0.46	0.15	0.01
stool	0.76	0.71	0.63	0.20	0.25	0.13	0.01
table	0.59	0.46	0.60	0.34	0.51	0.14	0.01
tent	0.52	0.53	0.51	0.29	0.21	0.10	0.01
toilet	0.53	0.49	0.48	0.23	0.11	0.08	0.01
tv stand	0.47	0.50	0.50	0.29	0.20	0.09	0.01
vase	0.61	0.65	0.64	0.36	0.34	0.12	0.02
wardrobe	0.65	0.63	0.61	0.36	0.43	0.10	0.02
xbox	0.46	0.57	0.47	0.28	0.25	0.06	0.01

References

1. Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: Robust & efficient point cloud registration using pointnet. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7163–7172 (2019) [5](#), [7](#), [9](#), [10](#), [12](#)
2. Chen, C., Li, G., Xu, R., Chen, T., Wang, M., Lin, L.: Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4994–5002 (2019) [5](#), [7](#)
3. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. *Image and Vision Computing* **10**(3), 145 – 155 (1992), range Image Understanding [5](#), [7](#), [9](#), [10](#), [12](#)
4. Eckart, B., Kim, K., Kautz, J.: Hgmr: Hierarchical gaussian mixtures for adaptive 3d registration. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 705–721 (2018) [5](#), [7](#), [9](#), [10](#), [12](#)
5. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **13**(4), 376–380 (1991) [4](#)
6. Wang, Y., Solomon, J.M.: Deep closest point: Learning representations for point cloud registration. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019) [5](#), [7](#), [9](#), [10](#), [12](#)
7. Wang, Y., Solomon, J.M.: Prnet: Self-supervised learning for partial-to-partial registration. In: Advances in Neural Information Processing Systems. pp. 8812–8824 (2019) [5](#), [7](#), [9](#), [10](#), [12](#)
8. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: European Conference on Computer Vision. pp. 766–782. Springer (2016) [5](#), [7](#), [9](#), [10](#), [12](#)