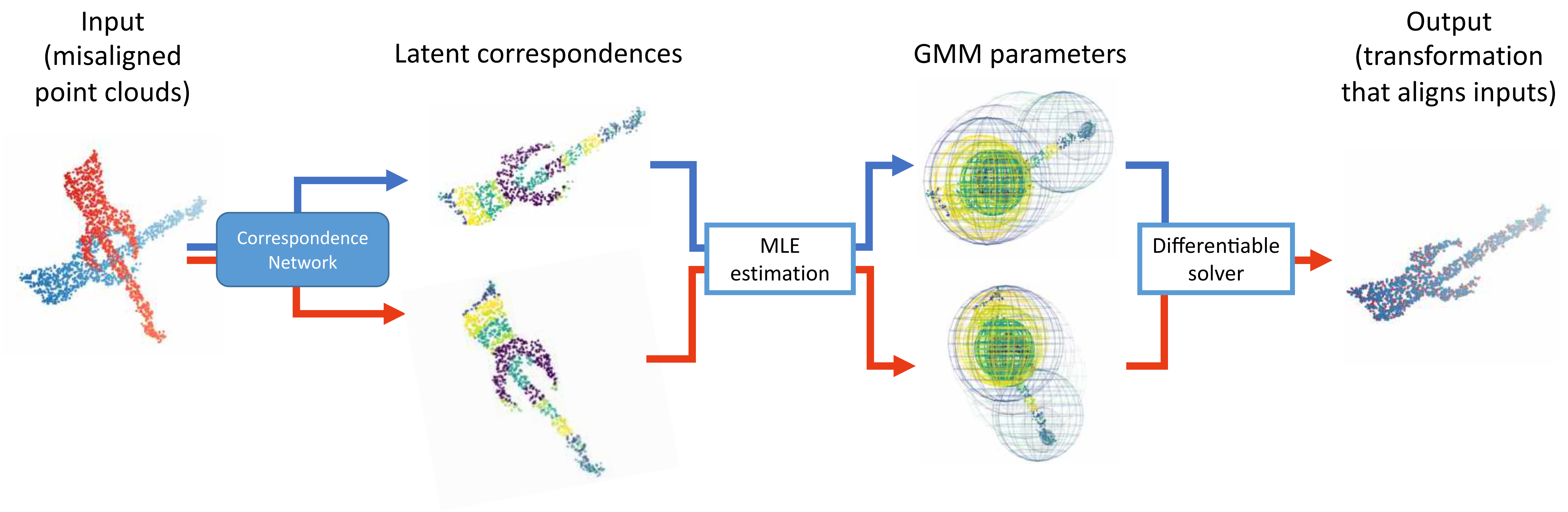
DeepGMR: Learning Latent Gaussian Mixture Models for Registration

Wentao Yuan Ben Eckart Kihwan Kim Varun Jampani Dieter Fox Jan Kautz






DeepGMR: Overview



Motivation

Limitations of Existing Registration Methods

Local: requires good initialization, e.g. ICP, HGMR

Inaccurate: requires further refinement, e.g. RANSAC10K

Inefficient: average run time > 1s for 1000 points, e.g. RANSAC10M

Not robust: accuracy drops on noisy data, e.g. DCP

Non-differentiable: cannot be used for gradient-based optimization, e.g. FGR

Properties of DeepGMR

Global: does not require pose or correspondence initialization

Accurate: outperforms state-of-the-art registration baselines

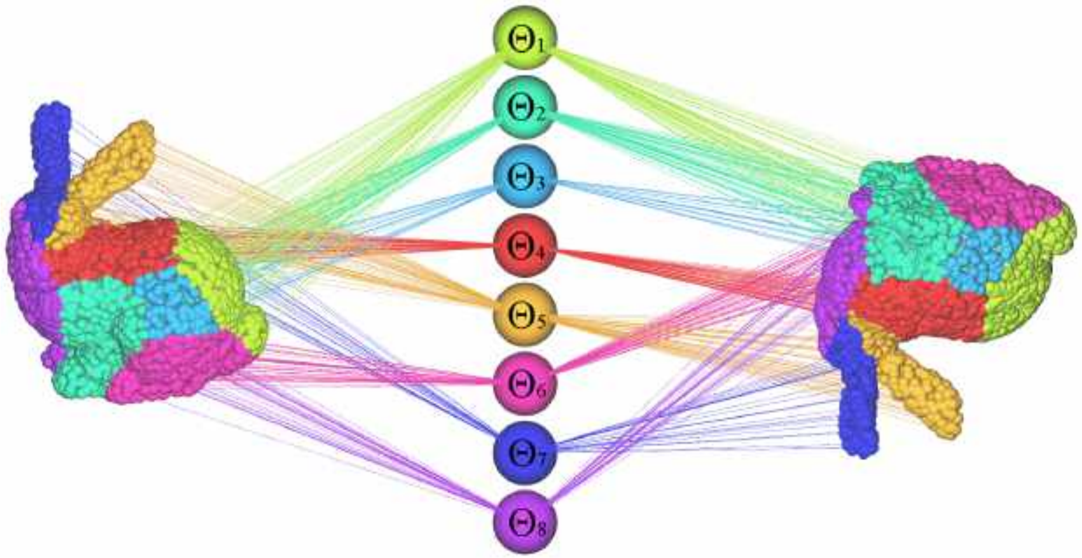
Efficient: average run time of 11 ms for point clouds with 1000 points

Robust: consistently good accuracy on noisy data

Differentiable: can be plugged into optimization that requires gradient

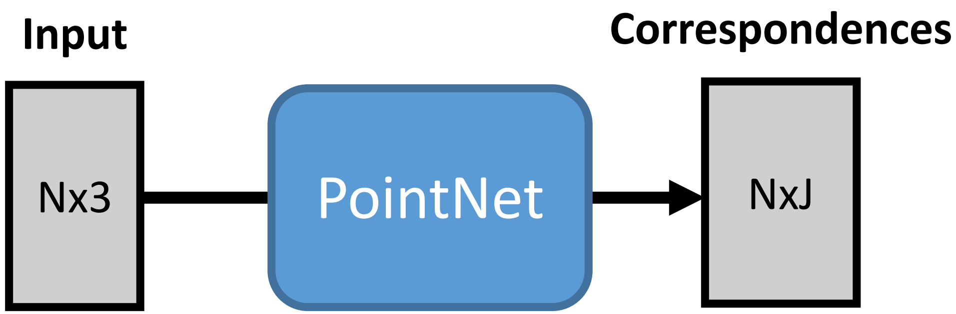
Method

Overview

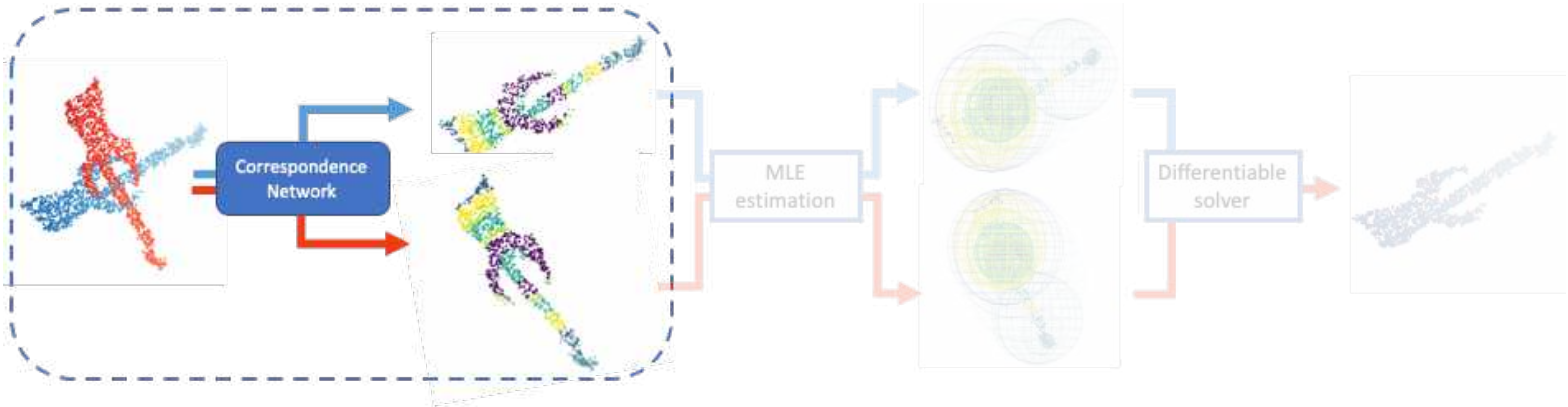


Input points → Correspondence Network → Correspondences → Differentiable Compute Blocks → Transformation

Correspondence Network



- Output $\mathbf{\Gamma} = \{\gamma_{ij}\}$: N x J correspondence matrix
- N: number of points, J: number of GMM components
- γ_{ij}: latent correspondence between **point *i*** and **component *j***



$\mathbf{M}_{\boldsymbol{\Theta}}$ Compute Block

Inputs

$\boldsymbol{\Gamma} = \{\gamma_{ij}\}$: N x J correspondence matrix

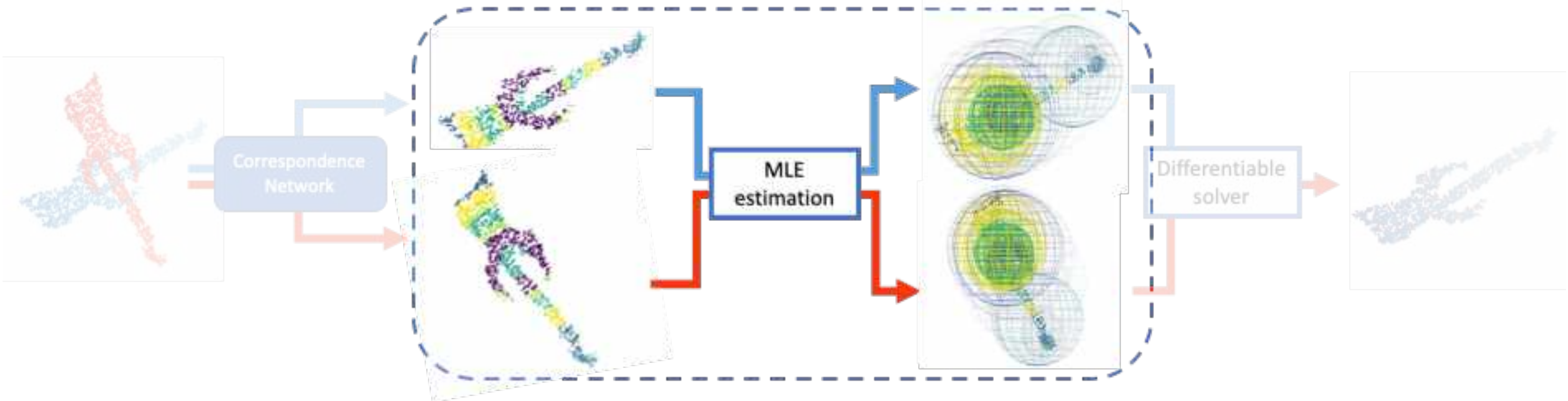
$\mathbf{P} = \{\boldsymbol{p}_i\}$: N x 3 point cloud

Outputs (GMM parameters $\boldsymbol{\Theta}$)

Weight: $\boldsymbol{\pi_j = \frac{1}{N}\sum_{i=0}^{N}\gamma_{ij}}$

Mean: $\boldsymbol{\mu_j = \sum_{i=0}^{N}\gamma_{ij}p_i}$

Covariance: $\boldsymbol{\Sigma_j = \sum_{i=0}^{N}\gamma_{ij}(p_i - \mu_j)(p_i - \mu_j)^T}$



$\mathbf{M_T}$ Compute Block

Inputs

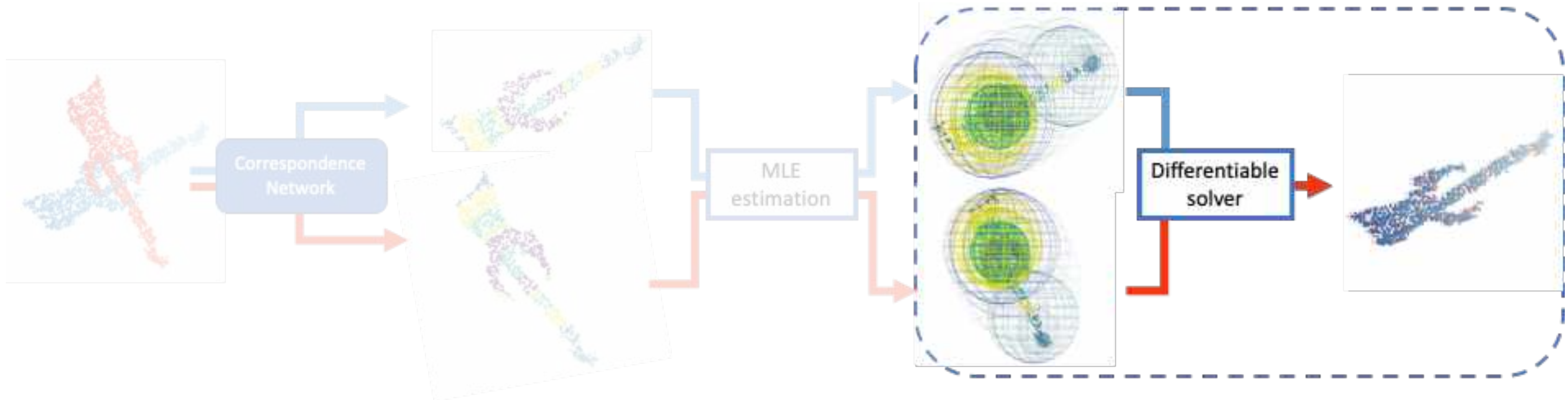
$\widehat{\boldsymbol{\Theta}} = \{\widehat{\boldsymbol{\pi}}_{\boldsymbol{j}}, \widehat{\boldsymbol{\mu}}_{\boldsymbol{j}}, \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{j}}\}$: source GMM

$\boldsymbol{\Theta} = \{\boldsymbol{\pi}_{\boldsymbol{j}}, \boldsymbol{\mu}_{\boldsymbol{j}}, \boldsymbol{\Sigma}_{\boldsymbol{j}}\}$: target GMM

Output (3D rigid transformation T^*)

$$T^* = \underset{T}{\operatorname{argmin}} \sum_{j=1}^{J} \widehat{\boldsymbol{\pi}}_{\boldsymbol{j}} \left\| T(\widehat{\boldsymbol{\mu}}_{\boldsymbol{j}}) - \boldsymbol{\mu}_{\boldsymbol{j}} \right\|_{\boldsymbol{\Sigma}_{\boldsymbol{j}}}$$

Complexity $O(J^3)$ with $J \ll N$

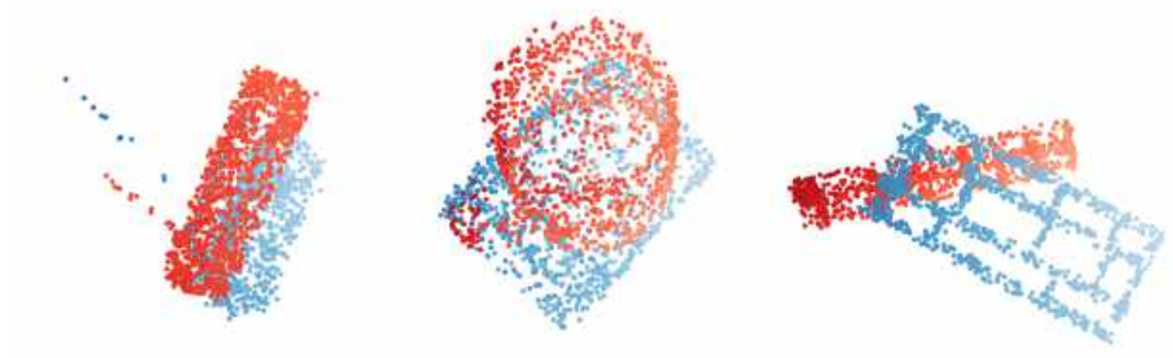


Results

Data

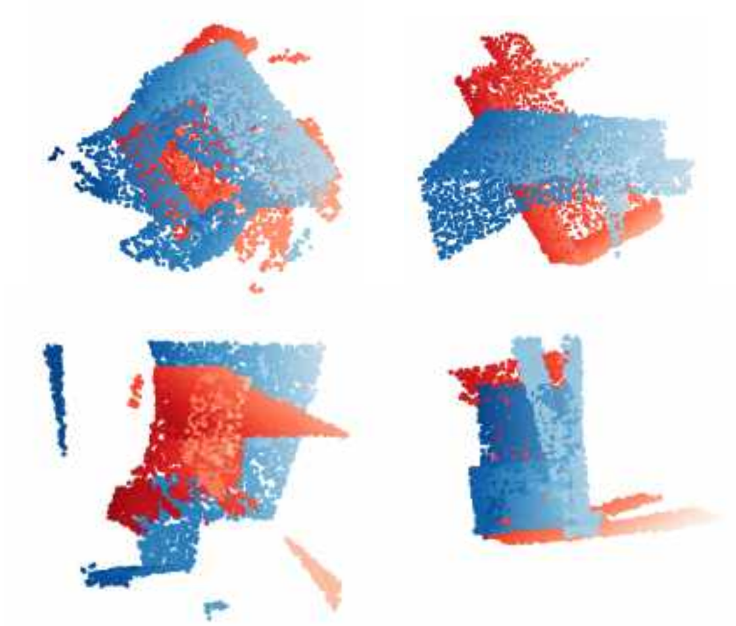
ModelNet (Chang et al. 2015)

- 12,311 models from 40 categories
- 3 variations
  - ModelNet Clean
  - ModelNet Noisy
  - ModelNet Unseen



ICL-NUIM (Choi et al. 2016)

- 1,478 scans from 4 rooms
- Real-world point clouds



Metrics

- RMSE

  T: predicted transformation T_{gt}: ground truth transformation   P = {p_i}: source point cloud

$$E_{RMSE} = \frac{1}{N}\sqrt{\sum_{i=1}^{n} \left\| T(p_i) - T_{gt}(p_i) \right\|^2}$$

- Recall (Re@0.2)
  - Percentage of test instances with $E_{RMSE} < 0.2$

Accuracy

Table 1. Average RMSE and recall with threshold 0.2 on various datasets. DeepGMR achieves the best performance across all datasets thanks to its ability to perform robust data association in challenging cases (Sec. 5.2). Local methods are labeled in red; Inefficient global methods are labeled in orange and efficient global methods (average runtime < 1s) are labeled in blue. Best viewed in color.

	ModelNet clean		ModelNet noisy		ModelNet unseen		ICL-NUIM	
	RMSE ↓	Re@0.2 ↑	RMSE ↓	Re@0.2 ↑	RMSE ↓	Re@0.2 ↑	RMSE ↓	Re@0.2 ↑
ICP [8]	0.53	0.41	0.53	0.41	0.59	0.32	1.16	0.27
HGMR [15]	0.52	0.44	0.52	0.45	0.54	0.43	0.72	0.50
PointNetLK [1]	0.51	0.44	0.56	0.38	0.68	0.13	1.29	0.08
PRNet [43]	0.30	0.64	0.34	0.57	0.58	0.30	1.32	0.15
RANSAC10M+ICP	0.01	0.99	0.04	0.96	0.03	0.98	0.08	0.98
TEASER++ [46]	**0.00**	**1.00**	**0.01**	**0.99**	**0.01**	**0.99**	0.09	0.95
RANSAC10K+ICP	0.08	0.91	0.42	0.49	0.30	0.67	0.17	0.84
FGR [48]	0.19	0.79	0.2	0.79	0.23	0.75	0.15	0.87
DCP [42]	0.02	0.99	0.08	0.94	0.34	0.54	0.64	0.16
DeepGMR	**0.00**	**1.00**	**0.01**	**0.99**	**0.01**	**0.99**	**0.07**	**0.99**

Efficiency

Table 2. Average running time (ms) of efficient registration methods on ModelNet40 test set. DeepGMR is significantly faster than other learning based method [1,42,43] and comparable to geometry-based methods designed for efficiency [15,48]. Baselines not listed (RANSAC10M+ICP, TEASER++) have running time on the order of 10s. OOM means a 16GB GPU is out of memory with a forward pass on a single instance.

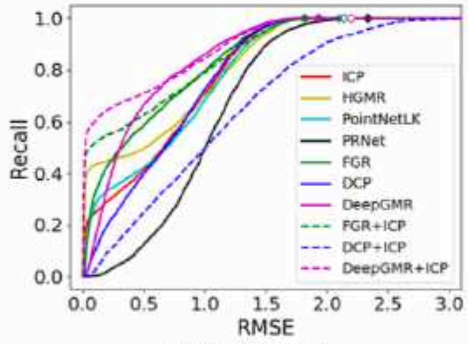
# points	ICP	HGMR	PointNetLK	PRNet	RANSAC10K+ICP	FGR	DCP	DeepGMR
1000	184	33	84	153	95	22	67	**11**
2000	195	35	90	188	101	32	90	**19**
3000	195	37	93	OOM	113	37	115	**26**
4000	198	39	106	OOM	120	40	135	**34**
5000	201	**42**	109	OOM	124	**42**	157	47

Limitation

- Assumes source and target point clouds are *i.i.d.* (independent and identically distributed) samples from the latent distribution
- Evaluation on partially overlapping point clouds

	RMSE ↓	Re@0.2 ↑
ICP [3]	2.45	0.29
HGMR [4]	0.58	0.45
PointNetLK [1]	0.66	0.33
PRNet [7]	0.79	0.12
FGR [8]	0.50	0.43
DCP [6]	0.68	0.19
DeepGMR	0.46	0.34
FGR+ICP	0.44	0.55
DCP+ICP	3.46	0.09
DeepGMR+ICP	**0.34**	**0.64**

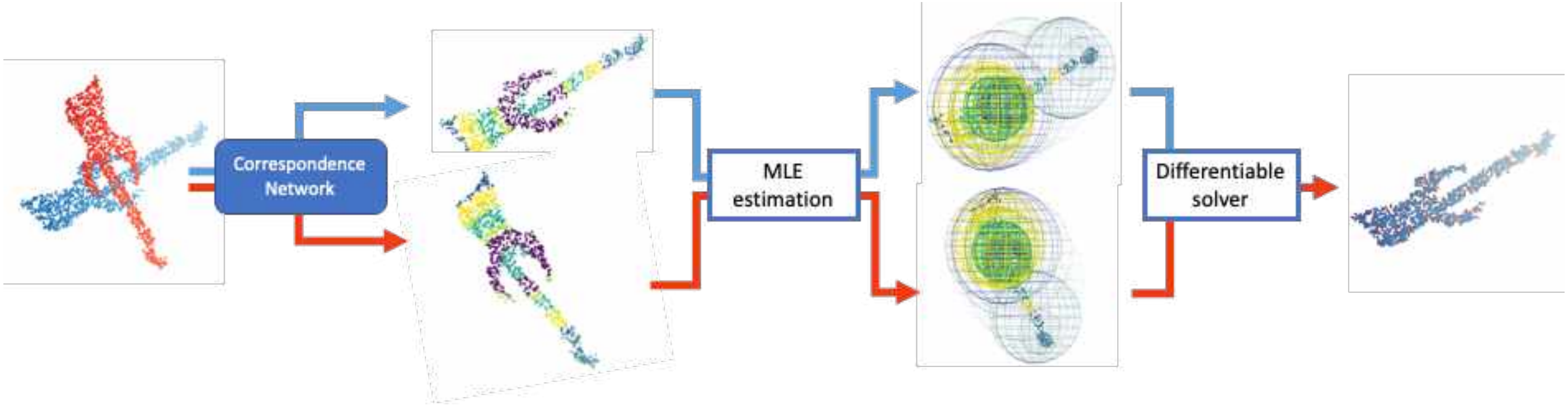
(a) RMSE and recall



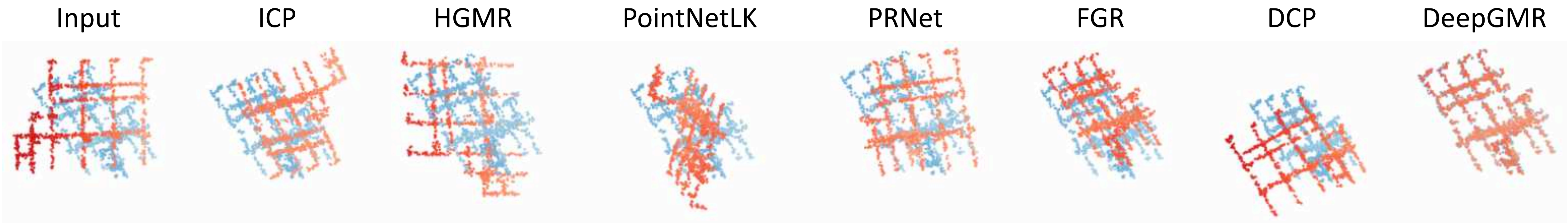
(b) CDF curves

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.

Conclusion



Correspondence Network
MLE estimation
Differentiable solver



Input
ICP
HGMR
PointNetLK
PRNet
FGR
DCP
DeepGMR