Appendix

A. Preliminaries

Equivariance. For a specified function $f : X \to Y$ as well as a specified group G, f is said to be equivariant with respect to a group action $g \in G$ if,

$$f(g \circ x) = g \circ f(x), \quad x \in X. \tag{9}$$

Vector Neuron. Since an SO(3)-equivariant operation *i.e.,* Vector Neuron (VN) [13] is utilized in our Point-wise Learner, here we provide a brief introduction of VN to make the paper self-contained.

The basic idea behind VN is to augment a scalar $z \in \mathbb{R}$ to a vector $\boldsymbol{v} \in \mathbb{R}^3$ to maintain SO(3) equivariance. For a point cloud $\mathcal{P}$, the Vector Neuron firstly formulates the raw representation $\mathcal{P} \in \mathbb{R}^{N\times 3}$ into a set of vector representations $\mathbf{V} \in \mathbb{R}^{N\times C\times 3}$ (the simplest condition is $C = 1$). Subsequently, it defines a linear mapping $f_{lin}(;\mathbf{W})$ acting on $\mathbf{V}$ to change the number of latent channels: $f_{lin}(\mathbf{V};\mathbf{W}) = \mathbf{W}\mathbf{V} \in \mathbb{R}^{N\times C'\times 3}$, where $\mathbf{W}$ represents the weight matrix $\mathbb{R}^{C'\times C}$. It can be demonstrated that such mapping is equivariant to arbitrary SO(3) rotations $\mathbf{R}$: $f_{lin}(\boldsymbol{r} \circ \mathbf{V};\mathbf{W}) = \mathbf{W}\mathbf{V}\boldsymbol{r}^{\mathrm{T}} = \boldsymbol{r} \circ f_{lin}(\mathbf{V};\mathbf{W})$, where $\boldsymbol{r}$ is a group action in $\mathbf{R}$.

Moreover, the nonlinear layer $f_{ReLU}(\cdot)$, pooling layer $f_p(\cdot)$ and normalization layer $f_n(\cdot)$ are redefined in VN. For more details, please refer to [13]. In this case, Eq. 2 in the main paper can be reformulated by:

$$\boldsymbol{v}_i^{l+1} = \mathbf{VN}(\boldsymbol{v}_j^l;\mathbf{W}) = f_p(f_{ReLU}(f_n(f_{lin}(\boldsymbol{v}_j^l;\mathbf{W})))). \tag{10}$$

Invariant Transformation. In Section 3.2, we adopt an invariant transformation [13] to generate the rotation-invariant signal $\mathbf{I}^P$. Specifically, for a set of equivariant vector representations $\mathbf{V} \in \mathbb{R}^{N\times C\times 3}$, a list of reference frames $\mathbf{F} \in \mathbb{R}^{N\times 3\times 3}$ can be calculated via multiple VN layers (*i.e.,* an equivariant mapping $\mathcal{M} : \mathbb{R}^{N\times C\times 3} \to \mathbb{R}^{N\times 3\times 3}$). The corresponding invariant signal is then produced by $\mathbf{I} = \mathbf{V}\mathbf{F}^{\mathrm{T}} = \mathbf{V}\mathcal{M}(\mathbf{V})^{\mathrm{T}} \in \mathbb{R}^{N\times C\times 3}$. Suppose a group action $\boldsymbol{r} \in$ SO(3) operating on $\mathbf{V}$, we have:

$$\begin{aligned}\mathbf{I}' &= (\boldsymbol{r} \circ \mathbf{V})\big(\mathcal{M}(\boldsymbol{r} \circ \mathbf{V})\big)^{\mathrm{T}} = (\mathbf{V}\boldsymbol{r}^{\mathrm{T}})\big(\mathcal{M}(\mathbf{V}\boldsymbol{r}^{\mathrm{T}})\big)^{\mathrm{T}} \\ &= \mathbf{V}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{r}\mathcal{M}(\mathbf{V})^{\mathrm{T}} = \mathbf{V}\mathbf{F}^{\mathrm{T}} = \mathbf{I}.\end{aligned} \tag{11}$$

It is obvious that this signal is invariant to arbitrary SO(3) rotations.

B. Theoretical Proof

Lemma 1. *The linear combinations of rotation-equivariant maps are still equivariant to rotations.*

Proof: Given two rotation-equivariant maps $\mathcal{M}_1 : X \to Y$, $\mathcal{M}_2 : X \to H$ and a group action $\boldsymbol{r}$ in SO(3), we have:

$$\begin{aligned}\boldsymbol{r} \circ \mathcal{M}_3(x) &= \boldsymbol{r} \circ \big(A\mathcal{M}_1(x) + B\mathcal{M}_2(x)\big) \\ &= A\big(\boldsymbol{r} \circ \mathcal{M}_1(x)\big) + B\big(\boldsymbol{r} \circ \mathcal{M}_2(x)\big) \\ &= A\mathcal{M}_1(\boldsymbol{r} \circ x) + B\mathcal{M}_2(\boldsymbol{r} \circ x) \\ &= \mathcal{M}_3(\boldsymbol{r} \circ x), \quad x \in X,\end{aligned} \tag{12}$$

where A, B are the linear coefficients. Based on Eq. 12, we can infer that the linear combinations of $\mathcal{M}_1$ and $\mathcal{M}_2$ are still equivariant to rotations.

Lemma 2. *The $\boldsymbol{v}_j^0 = [\boldsymbol{p}_{ji};\boldsymbol{n}_j;\boldsymbol{n}_j \times \boldsymbol{p}_{ji};\boldsymbol{c}_i]^T$ is equivariant to the rotation group SO(3) and invariant to the translation group.*

Proof: According to the geometrical relationship between points, we can know that $\boldsymbol{p}_{ji}$ is inherently equivariant to SO(3) and invariant to translations. Since $\boldsymbol{c}_i$ is the linear combination of $\boldsymbol{p}_{ji}$, we can conclude that $\boldsymbol{c}_i$ also have the same properties according to Lemma 1.

Given a covariance matrix $\boldsymbol{\Sigma} = \frac{1}{|N_i|}\sum_{\boldsymbol{p}_j \in N_i} \boldsymbol{p}_{ji}^{\mathrm{T}}\boldsymbol{p}_{ji}$, we get $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$ using the Singular Value Decomposition (SVD), where $\boldsymbol{\Lambda}$ is the 3×3 diagonal matrices of singular values, and $\mathbf{U}$ is the 3×3 eigenvectors. For the convenience of proof, we regard $\boldsymbol{\Sigma}$ as a mapping $\mathcal{M}_c : \mathbb{R}^{1\times 3} \to \mathbb{R}^{3\times 3}$. For a group action $\boldsymbol{r}$ in SO(3), we have:

$$\begin{aligned}\mathcal{M}_c(\boldsymbol{r} \circ \boldsymbol{p}_i) &= \frac{1}{|N_i|}\sum_{\boldsymbol{p}_j \in N_i} (\boldsymbol{r} \circ \boldsymbol{p}_{ji})^{\mathrm{T}}(\boldsymbol{r} \circ \boldsymbol{p}_{ji}) \\ &= \frac{1}{|N_i|}\sum_{\boldsymbol{p}_j \in N_i} (\boldsymbol{p}_{ji}\boldsymbol{r}^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{p}_{ji}\boldsymbol{r}^{\mathrm{T}}) \\ &= \frac{1}{|N_i|}\sum_{\boldsymbol{p}_j \in N_i} \boldsymbol{r}\boldsymbol{p}_{ji}^{\mathrm{T}}\boldsymbol{p}_{ji}\boldsymbol{r}^{\mathrm{T}} \\ &= \boldsymbol{r}\boldsymbol{\Sigma}\boldsymbol{r}^{\mathrm{T}} = \boldsymbol{r}\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}\boldsymbol{r}^{\mathrm{T}}.\end{aligned} \tag{13}$$

It can be seen that the eigenvector $\boldsymbol{n}_i$ is also rotated equally when $\boldsymbol{p}_i$ is rotated. Therefore, $\boldsymbol{n}_i$ is equivariant to rotations and invariant to translations. Since each element in $\boldsymbol{v}_j^0$ is equivariant to the rotation group SO(3) and invariant to the translation group, $\boldsymbol{v}_j^0$ also have the same properties.

C. Detailed Network Architecture

Equivariant Fully Convolutional Network. The proposed EFCN is based on the hierarchical architecture of KPConv. To ensure reproducibility, we provide the detailed network structure of our EFCN, as shown in Fig. 5. Note that, the "Attributes" represents extracting three geometrical attributes for each neighboring point to yield the $\boldsymbol{v}_j^0 \in \mathbb{R}^{4\times 3}$ in Sect. 3.2. By stacking multiple VN layers, a series of

Method	Voxelization						3D Cylinderical Convolutional Layers	Time
	N_p	J	K	L	δ	k_v	[kernel size, number of filters]	(ms)
SpinNet	2048	5	20	40	0.8	30	$[3 \times 3 \times 3, 32] \rightarrow [3 \times 3 \times 3, 64] \rightarrow [1 \times 3 \times 3, 64] \rightarrow [1 \times 3 \times 3, 128] \rightarrow [1 \times 3 \times 3, 128] \rightarrow [1 \times 3 \times 3, 256] \rightarrow [1 \times 3 \times 3, 256] \rightarrow [1 \times 2 \times 2, 32] \rightarrow [1 \times 2 \times 2, 32] \rightarrow [1 \times 2 \times 2, 32]$	0.75
Mini-SpinNet	512	3	7	20	0.8	10	$[3 \times 3 \times 3, 64] \rightarrow [1 \times 3 \times 3, 64] \rightarrow [1 \times 3 \times 3, 128] \rightarrow [1 \times 3 \times 3, 128] \rightarrow [1 \times 3 \times 3, 64] \rightarrow [1 \times 3 \times 3, 64] \rightarrow [1 \times 3 \times 3, 32] \rightarrow [1 \times 3 \times 3, 32]$	0.09

Table 8. The architecture discrepancies between SpinNet and Mini-SpinNet. N_p represents the number of sampling points in a local patch. J, K and L denote the number of divisions along the radial, elevation, azimuth dimension, respectively. δ and k_v represent the size of receptive field and the number of points in each voxel, respectively. Time indicates the consumption to compute per feature.

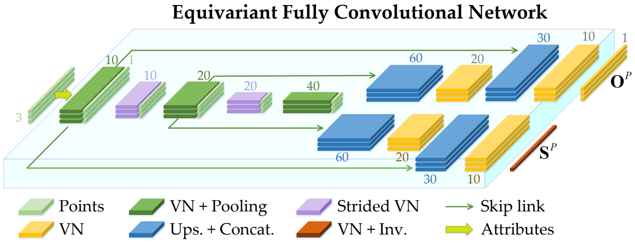

Figure 5. The detailed architecture of the proposed EFCN, where “Ups. + Concat.” represents the nearest upsampling followed by a concatenation.

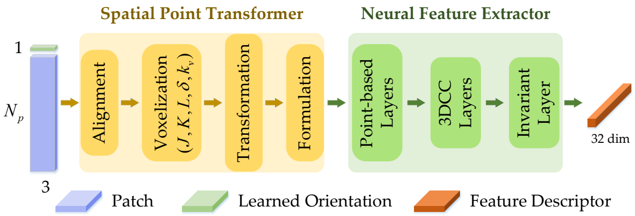

Figure 6. The detailed architecture of the Mini-SpinNet.

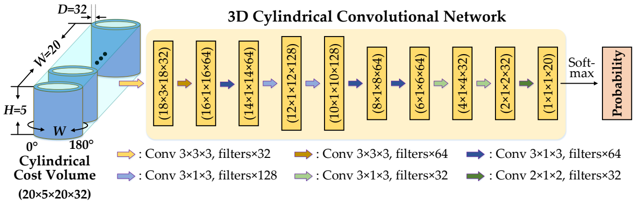

Figure 7. The detailed architecture of the 3DCCN.

equivariant features can be obtained progressively, further generating the rotation-invariant saliencies $\mathbf{S}^P$ and rotation-equivariant orientations $\mathbf{O}^P$. It can be seen that the whole network only leverages a smaller number of channels and simple steps to extract features. Therefore, our EFCN is lightweight and highly efficient.

Additionally, it also can be seen that the EFCN mainly consists of multiple VN layers, sampling/upsampling, and feature concatenation. The VN layer is inherently an equivariant map for the rotation group SO(3), and the sampling/upsampling and concatenation do not affect the rotational equivariance of features. Therefore, the EFCN is equivariant to SO(3) rotations.

Mini-SpinNet. To improve the efficiency and reduce memory usage, we adopt a lightweight patch-wise network, *i.e.,* Mini-SpinNet, to learn compact and general feature descriptors. The detailed architecture of Mini-SpinNet is depicted in Fig. 6 and the discrepancies between Spinnet and Mini-SpinNet are listed in Table 8. It can be observed that the Mini-SpinNet is responsible for encoding the input patch into a 32-dimensional feature descriptor. In particular, the Mini-SpinNet is nearly 9 times faster than the vanilla SpinNet by decreasing the hyperparameters in Voxelization and simplifing the 3DCC layers.

3D Cylindrical Convolutional Network. In the proposed Inliers Generator, we leverage a 3D Cylindrical Convolutional Network (3DCCN) to aggregate the cost, further obtaining a probability volume. The detailed architecture of 3DCCN is shown in Fig. 7. Unlike the 3DCC layers in Mini-SpinNet, each convolution in 3DCCN does not need to maintain the SO(2) equivariance. This is because the SO(2) rotation estimation is ready formulated into a permutation problem by building the cylindrical cost volume. Therefore, the 3DCCN actually degenerates into the 3DCNN in our Inliers Generator.

D. Details of Datasets

This section provides the details of the datasets used in the experiments. The main differences between the four datasets are shown in Fig. 9.

3DMatch [66] (training and test): This dataset contains a number of RGB-D frames, which consists of 62 real-world indoor scenes. We follow the official protocol in [66] to perform the training and test splits. All fragment pairs in 3DMatch have overlapping regions over 30%. We obtain 35,297 fragment pairs for training and 1,623 fragment pairs for testing.

3DLoMatch [26] (only test): This dataset can be regarded as a more challenging test set in the 3DMatch dataset. Different from the 3DMatch test set with over 30% overlaps, the 3DLoMatch only contains fragment pairs with overlaps between 10% and 30% and a total of 1,781 fragment pairs are selected for testing.

KITTI [20] (training and test): This dataset is com-

No.	Dataset	Acquisition	#Training	#Test	Type	Quality	Scale	Scenario
1	3DMatch [66]	RGBD camera	35,297	1,623	Real-World	Dense	Indoor	Room
2	3DLoMatch [26]	RGBD camera	-	1,781	Real-World	Dense	Indoor	Room
3	KITTI [20]	Velodyne-64 LiDAR sensor	1,358	555	Real-World	Sparse	Outdoor	Urban
4	ETH [49]	Hokuyo UTM-30LX Laser scanner	-	713	Real-World	Sparse	Outdoor	Street

Table 9. Datasets used in the evaluation.

posed of 11 sequences of outdoor scans acquired by Velodyne-64 3D LiDAR scanners. We follow the same dataset splits and preprocessing methods as used in [10]. Each pair of point cloud fragments is separated by at least 10m, where 1358 fragment pairs are used for training, 180 fragment pairs for validation, and 555 fragment pairs for testing.

ETH [49] (only test): This dataset is an outdoor street-level dataset captured by a Hokuyo UTM-30LX laser scanner. It consists of four scenes from different seasons and 713 point cloud fragments with overlaps larger than 30% are used for testing.

E. Implementation Details of BUFFER

Training. To make the whole network converge quickly, we first pre-train each module and then train the whole network. We apply random rotation augmentation on the target point cloud and then calculate the matched correspondences using the ground-truth transformation for training. For the Point-wise Learner, we exploit the same hyperparameters (such as voxel size and convolutional radius) as [5]. For the patch-wise Embedder, the support radius we use is same to [1]. The Adam optimizer with default parameters was used for network training. The learning rate is initially set to 0.001. We train each module for 10 epochs, halving the learning rate every 2 epochs.

Inference. On the ETH dataset, we employ 0.08m as the voxel size with a maximum point number of 30,000 to sample the raw scan. To select keypoints, a sigmoid function is first applied to the predicted point saliencies, and then those points with scores ≥ 0.5 are remained, finally 1500 points are randomly picked up from these remaining points as keypoints. For those datasets whose point clouds are gravity-aligned, we follow [1] to skip the alignment with a canonical orientation in the Patch-wise Embedder. To search inlier correspondences, we correlate the inlier distance threshold τ with keypoints $\boldsymbol{p}_i$, where $\tau = \|\boldsymbol{p}_i\| \pi / W$. In the hypothesis generation stage, we use RANSAC with 50,000 max iterations as well as default parameters.

F. Implementation Details of Baselines

We leverage the code and pre-trained models released by authors to conduct experiments. All baselines are implemented with PyTorch and run on the same hardware platform. Since the PyTorch implementation of D3Feat is unavailable on the KITTI dataset, we do not consider its running time in relevant experiments. For FCGF, D3Feat, Predator, YOHO, Gedi, and SpinNet, we find these methods can achieve better registration performance when more keypoints are adopted. To balance their efficiency and generalization, we follow [1, 5] to select 5000 keypoints for feature matching and use a RANSAC with 50,000 max iterations to estimate the transformation matrices.

G. Detailed Evaluation Metrics

Due to the discrepancies in scale and range between indoor and outdoor scenarios, we adopt different metrics to evaluate the registration quality on indoor and outdoor datasets.

Evaluation Metrics on 3DMatch and 3DLoMatch. On both indoor 3DMatch and 3DLoMatch datasets, we use the Registration Recall (RR) in [63] as evaluation metrics, which is defined as:

$$\text{RR} = \frac{1}{H}\sum_{h=1}^{H}\mathbb{1}\left(\sqrt{\frac{1}{N_c}\sum_{(\boldsymbol{p}_i,\boldsymbol{q}_i)\in\boldsymbol{\Omega}^*}\left\|\hat{\mathbf{R}}\boldsymbol{p}_i+\hat{\boldsymbol{t}}-\boldsymbol{q}_i\right\|^2}<\tau_r\right), \quad (14)$$

where H is the total number of fragments pairs, $\boldsymbol{\Omega}^* = \{\boldsymbol{p}_i, \boldsymbol{q}_i\}_{i=1...N_c}$ is a set of ground-truth point correspondences between $\mathcal{P}$ and $\mathcal{Q}$, and $\hat{\mathbf{T}} = \{\hat{\mathbf{R}}, \hat{\boldsymbol{t}}\}$ is the estimated rigid transformation. τ_r is the Mean Squared Error (MSE) threshold and set to 0.2m.

Evalution Metrics on KITTI and ETH. On both outdoor KITTI and ETH datasets, the Relative Translational Error (RTE), Relative Rotation Error (RRE), and Success rate are used as the evaluation metrics [41]. The RRE is defined as:

$$\text{RRE} = \arccos\left(\frac{\text{trace}(\hat{\mathbf{R}}^T\mathbf{R})-1}{2}\right)\frac{180}{\pi}, \quad (15)$$

Correspondingly, the RTE is defined as:

$$\text{RTE} = \left\|\hat{\boldsymbol{t}} - \boldsymbol{t}\right\|. \quad (16)$$

Here, $\hat{\mathbf{T}} = \{\hat{\mathbf{R}}, \hat{\boldsymbol{t}}\}$ and $\mathbf{T} = \{\mathbf{R}, \boldsymbol{t}\}$ represent the estimated and the ground-truth transformations, respectively. At last,

Method	*3DMatch*		*Generalized to ETH*	
	RR(%)↑	Time(s)↓	Success(%)↑	Time(s)↓
SpinNet	92.4	7.12	97.62	7.12
USIP [33]+SpinNet	85.4	0.90	77.42	0.78
D3Feat+SpinNet	91.6	2.28	90.18	2.40
PREDATOR+SpinNet	**93.0**	2.56	93.69	2.51
Ours	92.9	**0.20**	**99.30**	**0.26**

Table 10. Quantitative results of combining point-wise detector with the patch-wise descriptor, where SpinNet randomly samples 5,000 keypoints, USIP detects 512 keypoints, and others extract 1,500 keypoints.

Method	*3DMatch*		*Generalized to ETH*	
	RR(%)↑	Time(s)↓	Success(%)↑	Time(s)↓
with RANSAC [18]	**92.9**	**0.20**	99.30	**0.26**
with PointDSC [4]	92.2	0.23	**99.58**	0.30

Table 11. Quantitative results of our BUFFER with different outlier rejection methods.

Success rate can be calculated by:

$$\mathrm{SR} = \frac{1}{H}\sum_{h=1}^{H} \mathbb{1}\Big(\mathrm{RRE} < \tau_1 \text{ and } \mathrm{RTE} < \tau_2\Big). \qquad (17)$$

On the KITTI dataset, we adopt the setting of $\tau_1 = 1^\circ$ and $\tau_2 = 0.3m$. On the ETH dataset, τ_1 and τ_2 are set to 2° and $0.3m$, respectively.

H. Additional Quantitative Results

Point-wise Detector & Patch-wise Descriptor. In addition to combining point-wise and patch-wise networks like our BUFFER, another way is to directly connect the point-wise detector with the patch-wise descriptor. We combine a patch-wise descriptor with three point-wise detectors and conduct a series of comparative experiments to test the accuracy, efficiency, and generalizability of these methods. The quantitative results are shown in Fig. 10. It is found that the combination of detector and descriptor can achieve encouraging registration accuracy with fewer keypoints. This is primarily because the detected keypoints are more likely to be correctly matched. However, the simple combinations between detector and descriptor still suffer from low efficiency and inferior generalizability, which are difficult to meet practical requirements.

Ablation on Outlier Rejection. As explained in the main paper, our BUFFER does not contradict existing outlier rejection techniques and can also be combined with these methods to estimate an accurate rigid transformation. Here, we conduct two groups of experiments to investigate the impact of different outlier rejection modules on our BUFFER. Table 11 shows the quantitative results. We can find that the traditional method RANSAC achieves better accuracy and efficiency than the learned method PointDSC. This demonstrates that the RANSAC is effective for point correspondences with a high inlier rate. When being directly generalized to unseen domains, the PointDSC outperforms RANSAC marginally. This indicates that the our BUFFER could be further improved when combined with a better outlier rejection technique.

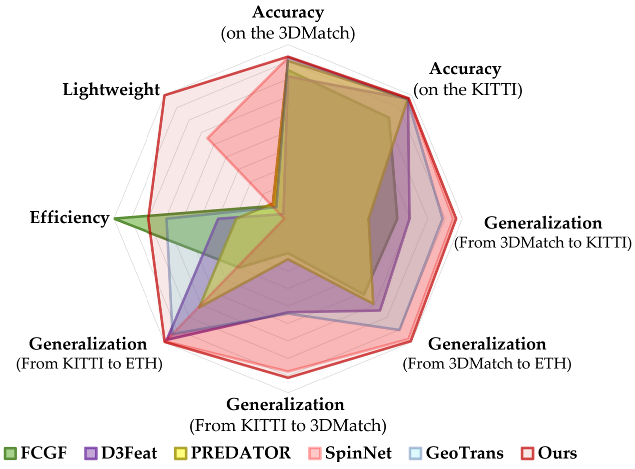

Figure 8. Comparison of the registration accuracy, generalization ability, efficiency, and lightweight of different methods.

I. Additional Qualitative Results

The quantitative results in Sect. 4 have demonstrated our BUFFER has superior accuracy, satisfactory efficiency, and strong generalization ability. In this section, we show more qualitative results.

Comprehensive Performance. To have an intuitive idea of the registration performance of different methods, we visualize the results in Sect. 4 as a "radar chart," as shown in Fig. 8. It is clear that our BUFFER is the most comprehensive registration method, achieving the best of both worlds in accuracy, efficiency, and generalization.

Generalization across unseen domains. The detailed qualitative results of generalization across unseen domains are shown in Fig. 9 and Fig. 10. Notice that the GeoTrans and SpinNet usually fail to align 3D scans when there are many planes or featureless regions such as floors and walls in indoor scenarios. Additionally, those geometrically-symmetric objects (*e.g.,* indoor desks and outdoor buildings) also cause the registration failure for GeoTrans and SpinNet. It also can be found that the scenes and data distribution in the 3DMatch dataset are significantly different from those in the KITTI dataset. Although these experiments are very challenging, the proposed BUFFER still achieves superior registration performance.

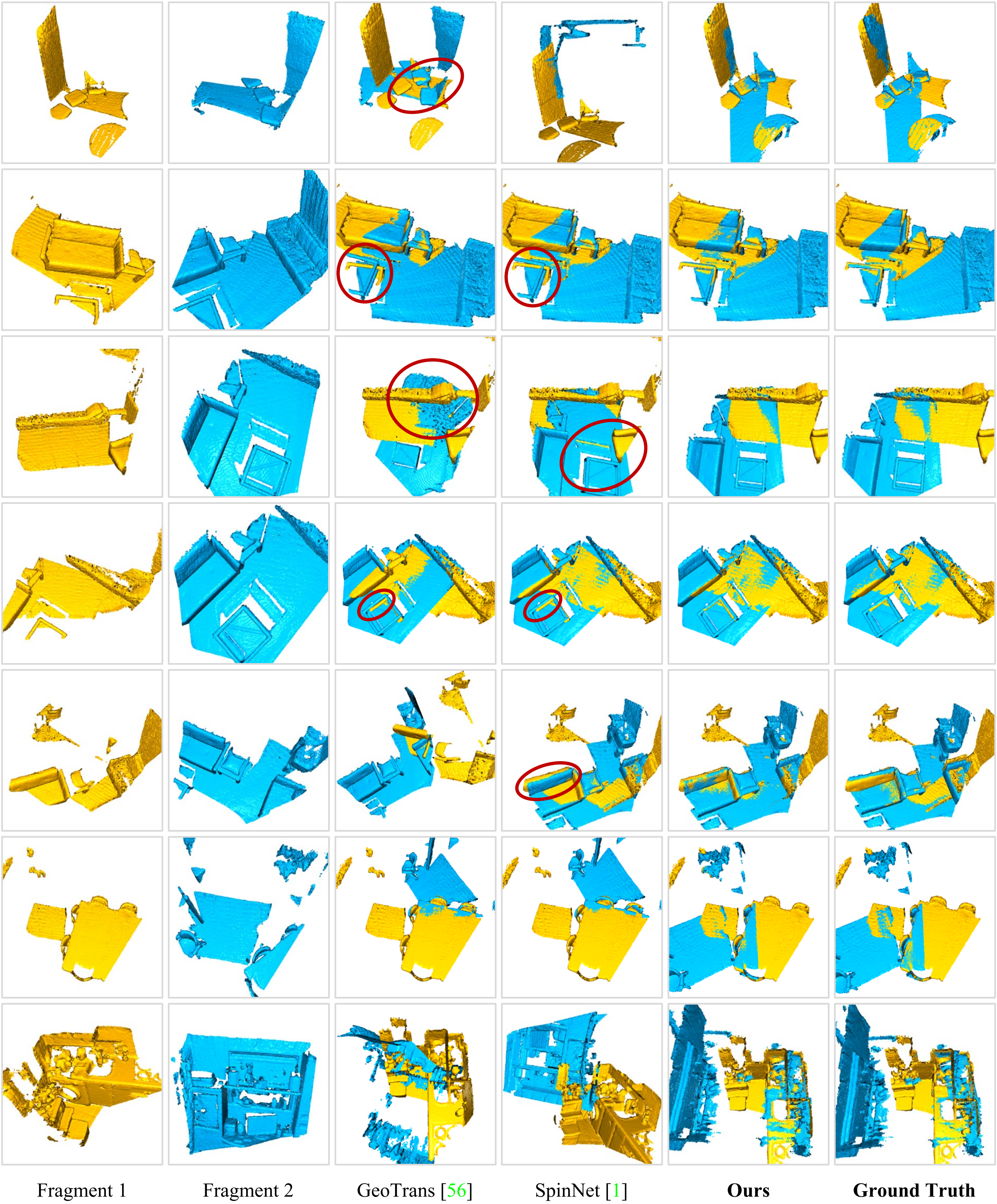

Figure 9. Qualitative results of generalization from outdoor KITTI to indoor 3DMatch.

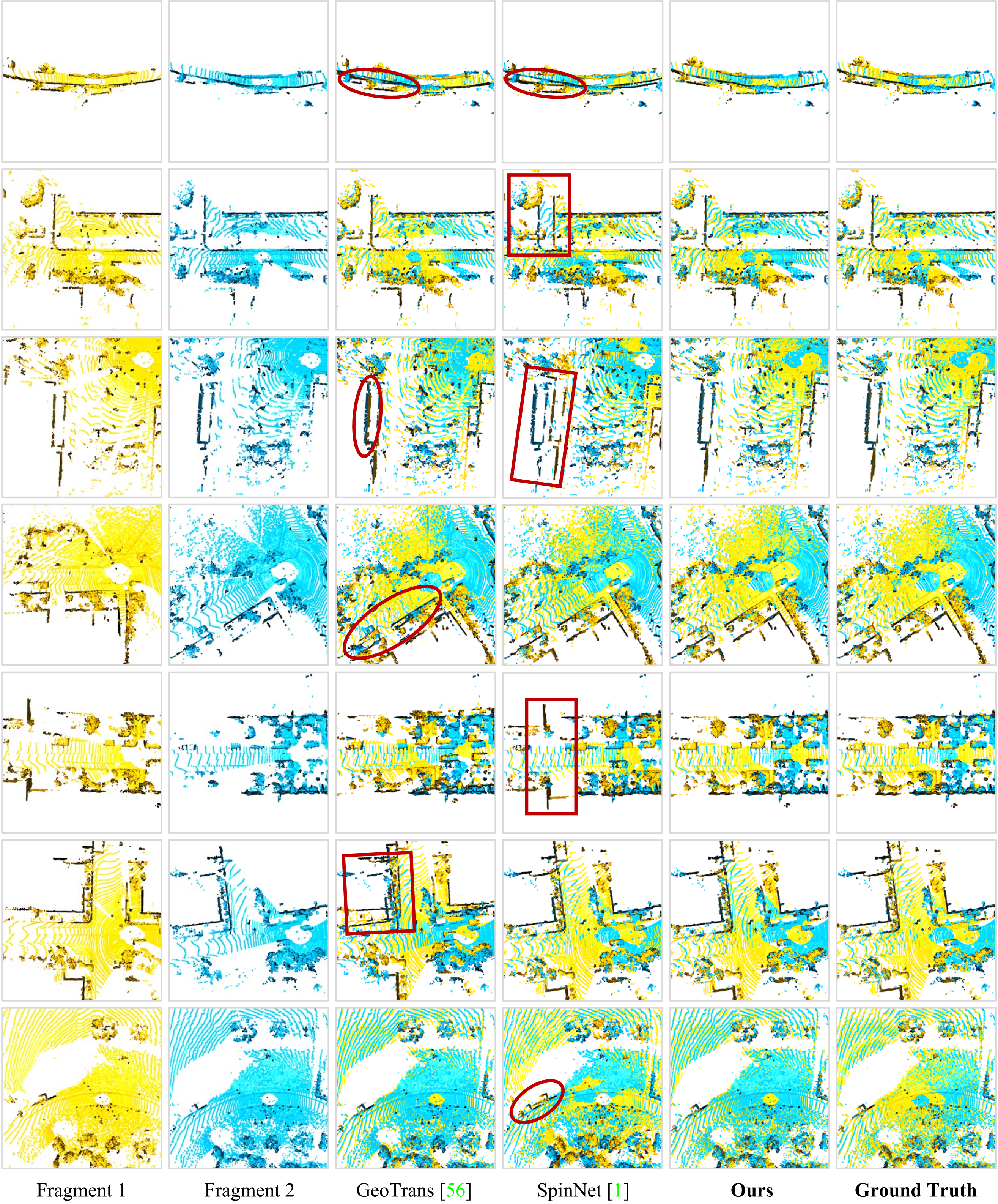

Figure 10. Qualitative results of generalization from indoor 3DMatch to outdoor KITTI.