Unlocking Generalization Power in LiDAR Point Cloud Registration

Supplementary Material

7. Method Details

Backbone. Our Point-Encoder and Point-Decoder follow the KPConv-FPN [32] structure to perform point-level feature extraction. Before inputting the points into the Point-Encoder, we voxel downsample the KITTI point cloud with a voxel size of 0.3m and the nuScenes point cloud with a voxel size of 0.2m. The Point-Encoder and Point-Decoder follow GeoTrans [28], with 5 and 3 layers, respectively. For the BEV-Encoder, we adopt a ResNet-like [15] structure with 3 layers.

BEV Patch and Superpoint Indexing. Given the superpoints, their 3D coordinates are projected onto the resolution $(H \times W)$ of the original BEV image. By accounting for the number of 2D max pooling operations β, we determine the index of each superpoint relative to the downsampled BEV patch. This correspondence establishes a one-to-one mapping between the features of the superpoints and BEV patches. Let (u_i, v_i) represent the 2D coordinates of a superpoint in the BEV image. The corresponding index in the downsampled BEV feature map is calculated as:

$$u\prime_i = \left\lfloor \frac{u_i}{2^\beta} \right\rfloor, \quad v\prime_i = \left\lfloor \frac{v_i}{2^\beta} \right\rfloor. \tag{6}$$

Here, (u_i', v_i') represents the index of the patch feature in the downsampled BEV image.

Point Matching. After obtaining the superpoint correspondences $\hat{\mathcal{C}}$, we follow a point-to-node assignment strategy [28] to uniquely assign dense points $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{Q}}$ to their nearest superpoints, resulting in groups $\mathcal{G}^P$ and $\mathcal{G}^Q$. Then, based on the superpoint correspondences $\hat{\mathcal{C}}$, we perform local dense matching. Unlike previous work, we input the attention features $\bar{\mathbf{H}}^{\hat{P}}$ and $\bar{\mathbf{H}}^{\hat{Q}}$ into the Point-Decoder to obtain dense point features $\mathbf{F}_{\tilde{P}} \in \mathbb{R}^{\tilde{n}\times\tilde{d}}$ and $\mathbf{F}_{\tilde{Q}} \in \mathbb{R}^{\tilde{m}\times\tilde{d}}$. For a given superpoint correspondence $\hat{\mathcal{C}}_i$ and its corresponding local dense points $\mathcal{G}_i^P$ and $\mathcal{G}_i^Q$, we compute the cost matrix $\tilde{\mathbf{C}}$, where $\tilde{\mathbf{C}}_i = \mathbf{F}^i_{\tilde{P}} \left(\mathbf{F}^i_{\tilde{Q}}\right)^T / \sqrt{\tilde{d}}$. Then, we use the Sinkhorn algorithm [30] to recompute the similarity matrix, resulting in $\bar{\mathbf{C}}$. Based on $\bar{\mathbf{C}}$, we apply mutual top-k to select the dense point correspondences. Finally, we gather all the dense point correspondences $\tilde{\mathcal{C}}_i$ for each coarse match in $\hat{\mathcal{C}}$, forming the final dense point correspondences $\mathcal{C} = \bigcup_{i=1}^{|\hat{\mathcal{C}}|} \tilde{\mathcal{C}}_i$.

Loss Function. Our framework's loss function consists of two components, $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_f$, where L_c and L_f represent the same superpoint matching loss and point matching loss as [28].

Implementation Details. We conduct our experiments using PyTorch [26] on an Intel (R) Xeon (R) Gold 5118 CPU and an NVIDIA RTX 3090 GPU. The Adam optimizer [17] is used to train our model, with an initial learning rate of 1e-4 and a weight decay of 1e-6.

8. Evaluation Metrics

We follow previous work [28] and report Patch Inlier Ratio (PIR), Inlier Ratio (IR), Relative Rotation Error (RRE), Relative Translation Error (RTE) and Registration Recall (RR).

Patch Inlier Ratio (PIR) represents the proportion of superpoint (patch) matches that correctly overlap when aligned using the ground-truth transformation $\mathbf{T}_{\mathbf{P}\to\mathbf{Q}}$. This metric indicates the reliability and accuracy of the proposed superpoint (patch) correspondences:

$$\mathrm{PIR} = \frac{1}{|\hat{\mathcal{C}}|} \sum_{(\hat{p}_i,\hat{q}_j)\in\hat{\mathcal{C}}} \mathbb{1}(\exists \tilde{\mathbf{p}} \in \mathcal{G}_i^P, \tilde{\mathbf{q}} \in \mathcal{G}_i^Q \text{ s.t. } \|\mathbf{T}_{\mathbf{P}\to\mathbf{Q}}(\tilde{\mathbf{p}}) - \tilde{\mathbf{q}}\|_2 < \tau), \tag{7}$$

where $\tau = 0.6$m and $\mathbb{1}$ is the indicator function.

Inlier Ratio (IR) represents the proportion of inlier matches among all candidate point correspondences. A match qualifies as an inlier if the distance between the two points transformed by the ground-truth transformation $\mathbf{T}_{\mathbf{P}\to\mathbf{Q}}$ is less than a threshold τ_1 = 1.0m:

$$\mathrm{IR} = \frac{1}{|\mathcal{C}|} \sum_{(\tilde{p}_i,\tilde{q}_j)\in\mathcal{C}} \mathbb{1}\left(\left\|\mathbf{T}_{\mathbf{P}\to\mathbf{Q}}\left(\tilde{\mathbf{p}}_i\right) - \tilde{\mathbf{q}}_i\right\|_2 < \tau_1\right). \tag{8}$$

Relative Rotation Error (RRE) represents the geodesic distance measured in degrees between the estimated $\mathbf{R}_{est}$ and ground-truth $\mathbf{R}_{gt}$ rotation matrices. It quantifies the discrepancy between the predicted and actual rotation matrices:

$$\mathrm{RRE} = \arccos\left(\frac{\operatorname{trace}\left(\mathbf{R}_{est}^T \cdot \mathbf{R}_{gt} - 1\right)}{2}\right). \tag{9}$$

Relative Translation Error (RTE) represents the Euclidean distance between the estimated $\mathbf{t}_{est}$ and ground-truth $\mathbf{t}_{gt}$ translation vectors. This metric assesses the difference between the estimated and ground-truth translation vectors:

$$\mathrm{RTE} = \|\mathbf{t}_{est} - \mathbf{t}_{gt}\|_2. \tag{10}$$

Registration Recall (RR) for all outdoor datasets is defined as the proportion of point cloud pairs where both RRE and RTE fall below specified thresholds (RRE $< 5°$ and RTE < 2m):

$$\mathrm{RR} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}\left(\mathrm{RRE}_i < 5° \wedge \mathrm{RTE}_i < 2\mathrm{m}\right), \tag{11}$$

Model	**KITTI@10m**					**KITTI@20m**					**KITTI@30m**					**Train on KITTI@40m**					
	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	mRR
FCGF [8]	1.180	0.285	6.713	3.304	65.6	1.846	0.532	15.044	18.343	5.0	–	–	21.267	29.356	0.0	2.089	0.524	23.732	39.583	0.7	17.8
Predator [16]	1.560	1.073	7.622	9.567	0.4	–	–	14.587	19.495	0.0	1.873	0.607	18.341	29.323	0.5	2.037	1.165	22.941	38.708	0.7	0.4
CoFiNet [39]	1.325	0.280	7.400	3.420	61.4	1.899	0.588	23.895	17.138	8.5	1.897	0.678	28.712	24.868	10.3	2.375	0.888	23.776	25.417	23.7	26.0
GeoTrans [28]	1.296	0.300	38.783	19.247	42.2	1.133	0.319	16.224	5.267	68.7	1.183	0.375	**5.524**	**2.217**	**80.5**	**1.037**	**0.514**	**8.639**	**3.235**	**85.6**	69.3
BUFFER [2]	1.119	0.182	8.679	1.893	81.6	2.202	0.435	36.861	12.615	30.2	2.747	0.556	70.534	27.838	3.2	–	–	76.648	39.030	0.0	28.8
PARE [38]	4.022	1.383	62.691	43.978	0.4	0.951	0.293	63.597	23.863	0.4	3.275	**0.329**	66.382	18.167	1.1	1.695	0.685	46.054	16.752	25.2	27.1
UGP (Ours)	**0.384**	**0.117**	**0.749**	**0.263**	**98.6**	**0.794**	**0.249**	**3.652**	**1.613**	**89.7**	**1.139**	0.378	13.235	5.616	78.9	1.385	0.596	17.767	8.927	71.9	**84.8**

Table 6. **Cross-distance generalization experiments. We train at KITTI@40m** and then test at 40m and nearer distances at 10m, 20m, and 30m. RRE and RTE denote the error for successfully matched point cloud pairs, while RRE* and RTE* reflect the error for all point cloud pairs, providing a more comprehensive evaluation. The final column shows the mean Registration Recall.

Model	**Train on KITTI@10m**					**KITTI@20m**					**KITTI@30m**					**KITTI@40m**					
	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	mRR
CoFiNet [39]	0.699	0.175	1.912	0.731	94.2	1.739	0.488	9.901	8.917	46.6	1.934	0.878	22.910	28.894	1.1	-	-	24.768	39.194	0.0	35.5
GeoTrans [28]	**0.291**	**0.082**	0.358	**0.095**	99.3	2.453	0.815	36.227	26.501	2.1	4.298	1.232	44.197	35.805	0.5	2.229	0.855	48.724	45.975	1.4	25.8
BUFFER [2]	0.309	0.091	**0.311**	0.097	**99.8**	0.645	0.188	2.988	1.554	92.5	0.997	**0.291**	24.145	15.086	51.4	1.511	**0.445**	45.820	30.337	20.1	66.0
UGP (Ours)	0.296	0.093	0.336	0.110	99.5	**0.488**	**0.170**	**0.615**	**0.274**	**97.5**	**0.816**	0.354	**5.348**	**1.879**	**90.3**	**1.091**	0.527	**19.566**	**11.560**	**66.9**	**88.6**

Table 7. **Cross-distance generalization experiments on KITTI-Sparse. We train at KITTI@10m** and then **test at KITTI-Sparse@10m** and farther distances at **KITTI-Sparse@20m, KITTI-Sparse@30m, and KITTI-Sparse@40m**. KITTI-Sparse denotes that we use farthest point sampling (FPS) to downsample the input point clouds to **5000** points. RRE and RTE denote the error for successfully matched point cloud pairs, while RRE* and RTE* reflect the error for all point cloud pairs, providing a more comprehensive evaluation. The final column shows the mean Registration Recall.

where M is the number of point cloud pairs to be aligned.

9. Additional Experiments

9.1. Cross-distance (train on KITTI@40m)

To comprehensively evaluate the generalization across different distances, we train on long-distance data (KITTI@40m) and generalize to shorter distances, as shown in Tab. 6. The results reveal that methods such as FCGF [8], Predator [16], CoFiNet [39], BUFFER [2], and PARE [38] struggle to achieve direct convergence at long distances. In contrast, the GeoTrans [28] network demonstrates the ability to converge at long distances and deliver good performance. However, its performance drops significantly when applied to simpler scenarios, such as KITTI@10m and KITTI@20m. This suggests that GeoTrans heavily relies on the visible data distribution, further highlighting that its cross-attention mechanism fails to adapt to variations in consistency representation of the same structure across different distances and datasets. Consequently, it cannot learn robust and generalizable features for LiDAR scenes. In contrast, our method not only successfully converges on KITTI@40m, but also gradually improves its performance as the distance decreases, consistent with the expected difficulty of the registration task. Ultimately, our method UGP achieves an mRR of 84.8%, which is 15.5% significantly ahead of the suboptimal GeoTrans.

9.2. Cross-dataset (KITTI@10m to Waymo@10m)

To comprehensively evaluate the cross-dataset generalization ability of our method, we supplemented the results with training on KITTI@10m and testing on Waymo@10m. For the Waymo dataset (64-line LiDAR), we follow the same protocol in [31] and utilize the testing subset, the results are shown in Tab. 8. Our method achieved a state-of-the-art RR of 100% and the lowest errors in both RRE and RTE.

Model	**Waymo@10m**				
	RRE(°)	RTE(m)	RRE*(°)	RTE*(m)	RR(%)
FCGF [8]	**0.137**	0.081	0.597	0.155	99.2
SpinNet [1]	0.377	0.096	0.553	0.171	99.2
Predator [16]	0.190	0.082	0.190	0.082	**100.0**
CoFiNet [39]	0.179	0.080	0.179	0.080	**100.0**
GeoTrans [28]	0.255	0.124	3.740	7.247	61.5
Buffer [2]	0.171	0.088	0.171	0.088	**100.0**
PARE [38]	0.270	0.136	1.051	4.003	76.9
UGP (Ours)	**0.137**	**0.075**	**0.137**	**0.075**	**100.0**

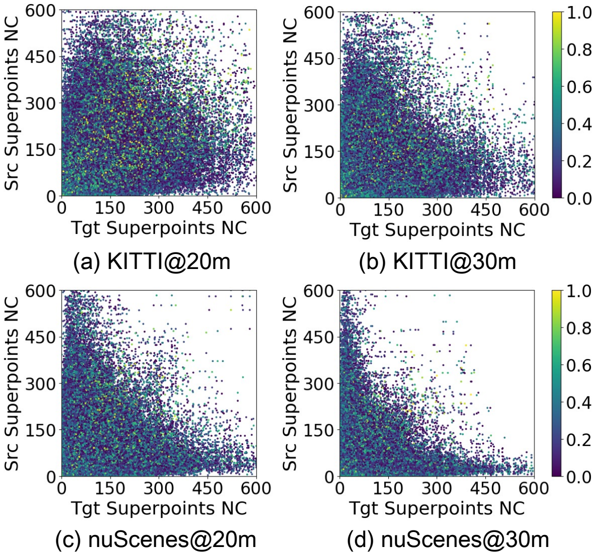
Table 8. The results of the cross-dataset generalization experiments **from KITTI@10m to Waymo@10m**.

9.3. KITTI-Sparse

To evaluate the robustness of the network to extremely sparse LiDAR point clouds, we use farthest point sampling (FPS) to downsample the input point clouds to **5000** points, which is referred to as KITTI-Sparse. We compare CoFiNet [39], GeoTrans [28], and BUFFER [2], with the results shown in Tab. 7. UGP demonstrates a significant advantage at 20m, 30m, and 40m. Compared to the suboptimal BUFFER, UGP achieves 97.5% (+5.0%) RR at 20m, 90.3% (+38.9%) RR at 30m, and 66.9% (+46.8%) RR at 40m.

		Train on KITTI@10m					KITTI@20m					KITTI@30m					KITTI@40m					
Noise σ	**Method**	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	RRE	RTE	RRE*	RTE*	RR	mRR
0.01	FCGF [8]	**0.214**	**0.061**	0.868	0.153	98.9	**0.389**	**0.128**	5.507	1.323	92.8	1.080	0.415	11.739	10.998	62.0	1.634	0.835	20.555	30.727	18.7	68.1
	BUFFER [2]	0.266	0.073	**0.269**	0.079	**99.8**	0.472	**0.128**	0.520	0.280	98.6	0.669	**0.237**	**1.366**	1.876	93.0	1.037	**0.363**	13.327	15.820	61.1	88.1
	UGP (ours)	0.243	0.071	0.294	**0.078**	**99.8**	0.399	0.144	**0.441**	**0.219**	**99.3**	**0.641**	0.282	2.206	**1.052**	**95.7**	**0.994**	0.478	**12.938**	**7.174**	**78.4**	**93.3**
0.03	FCGF [8]	**0.204**	**0.060**	0.908	0.158	98.7	**0.374**	**0.117**	6.149	1.731	91.4	1.014	0.396	11.600	11.886	58.2	1.648	0.728	22.072	31.061	17.9	66.6
	BUFFER [2]	0.277	0.074	**0.278**	0.080	**99.8**	0.478	0.127	0.508	0.342	98.6	0.720	**0.243**	3.293	2.026	93.0	0.959	**0.358**	18.489	17.286	55.4	86.7
	UGP (ours)	0.246	0.070	0.292	**0.077**	**99.8**	0.419	0.147	**0.461**	**0.222**	**99.3**	**0.607**	0.285	**2.054**	**1.180**	**95.7**	**0.880**	0.451	**16.181**	**6.449**	**77.0**	**93.0**
0.05	FCGF [8]	**0.216**	**0.061**	1.205	0.235	98.6	**0.422**	0.130	5.774	1.839	91.0	1.248	0.410	11.973	14.004	53.3	1.601	1.004	19.855	31.803	17.0	65.0
	BUFFER [2]	0.287	0.073	0.289	0.080	**99.8**	0.496	**0.129**	0.544	0.280	98.6	0.790	**0.252**	**3.491**	2.286	91.9	1.119	**0.397**	18.584	17.225	51.8	85.5
	UGP (ours)	0.253	0.070	**0.285**	**0.077**	**99.8**	0.427	0.150	**0.460**	**0.224**	**99.3**	**0.597**	0.283	3.554	**1.214**	**93.5**	**1.101**	0.488	**14.526**	**8.571**	**75.5**	**92.0**

Table 9. **Comparison of results under varying noise intensities**, with σ representing the standard deviation. RRE and RTE denote the error for successfully matched point cloud pairs, while RRE* and RTE* reflect the error for all point cloud pairs, providing a more comprehensive evaluation. The final column shows the mean Registration Recall.



(a) KITTI@20m

(b) KITTI@30m

(c) nuScenes@20m

(d) nuScenes@30m

Figure 7. **Visualization of data distributions for ground truth matching point pairs across varying distances and datasets**. (a-d) Each point in the figure represents a ground truth corresponding superpoint pair. The position of each point indicates the neighborhood count (NC) of the superpoint within a radius of $r = 2.4$m in both the source (src) and target (tgt) point clouds, and the color represents the overlap degree of the corresponding superpoint pairs after rotation by the ground truth transformation.

9.4. KITTI-Noise

To evaluate the robustness of our method to noise in real-world environments, we add Gaussian-distributed random noise $N\left(0, \sigma^2\right)$ (clipped to $[-3\sigma, +3\sigma]$) to each point's position to simulate the measurement errors and noise encountered by LiDAR sensors in real-world scenarios, as shown in Tab. 9. Since other methods almost completely fail at long ranges, we only compare our method with FCGF [8] and BUFFER [2]. Our method achieves the highest RR across different levels of noise. Notably, the RR of our method remains unaffected at short distances, such as KITTI@10m and KITTI@20m. At KITTI@30m, the RR decreases by 3.3% under a noise level of $\sigma = 0.05$. At KITTI@40m, the decrease reaches 6.5% under the same noise level. In summary, our method experiences an mRR reduction of no more than 2.5% (from 94.5% to 92.0%) at an intensity of $\sigma = 0.05$, demonstrating a certain level of robustness to noise.

LayerNum	KITTI@30m (RR%)	KITTI@40m (RR%)
2	95.1	80.6
3	**96.8**	**82.0**
4	95.7	74.8
5	96.2	72.7
6	95.7	72.7

Table 10. **Ablation experiment** of progressive self-attention module partitioning **with different number of spatial layers** L. $L = 3$ is selected to achieve the highest RR on KITTI@30m and KITTI@40m.

9.5. Ablation of Parameter

We conducted an ablation study on the number of layers L used to divide the space in the PSA. As shown in Tab. 10, we selected $L = 3$, which provided the best performance, as the final network implementation.

9.6. Mechanism Analysis

Admittedly, cross-attention achieves promising performance under same-distance/dataset settings. However, for LiDAR registration requiring cross-domain generalization, we identify a fundamental limitation: Cross-attention learns static density matching patterns from training data but struggles to extrapolate to the real physical law of LiDAR density decay ($\rho \propto 1/d^2$). When applied to cross-distance or cross-dataset scenarios, the learned correlation patterns become invalid due to density scaling or differences in LiDAR type.

To this end, we analyzed the $\mathrm{Softmax}(QK^T/\sqrt{d})$ mechanism in UGP w. cross-attention, as shown in Fig. 8. At K@10m training, regions with high cross-attention scores (Top-10) covered 91.64% of true matches, *validating the effectiveness of the QK^T mechanism in capturing* *point cloud correspondences and guiding feature updates*. However, cross-distance or cross-dataset scenarios exhibit LiDAR distribution shifts, causing true matches in high-score regions to plummet. This introduces false matches, increases feature ambiguity, and weakens generalization.

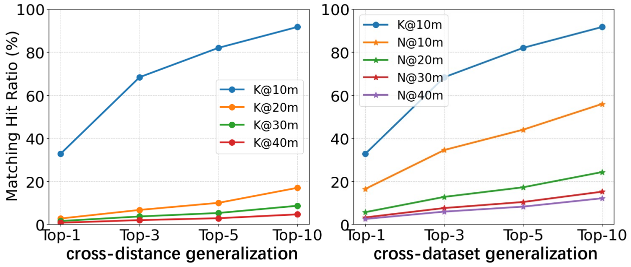


Figure 8. **Visualization of the matching hit ratio** in cross-distance and cross-dataset generalization experiments.

10. Visualizations

LiDAR Point Cloud Registration Characteristics. To supplement Fig. 2 (a) in Sec. 3, we provide additional details. Specifically, Fig. 7 illustrates the data distributions of ground truth matching point pairs for KITTI and nuScenes at distances of 20m and 30m.

Registration Results. The cross-distance registration results for KITTI and nuScenes are shown in Fig. 9 and Fig. 10.

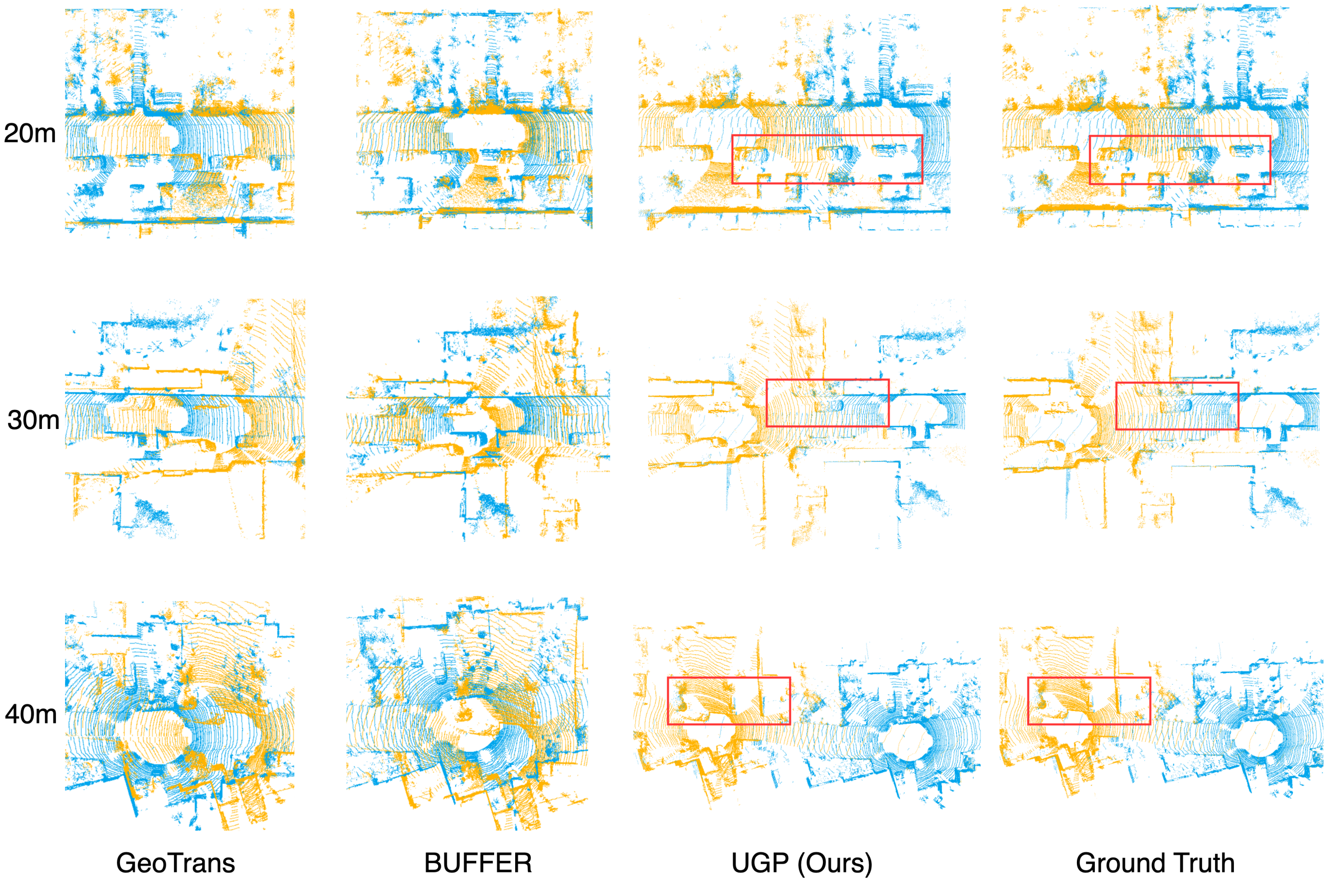


Figure 9. **Cross-distance generalization visualization** of GeoTrans [28], BUFFER [2], and UGP **on the KITTI [12]** dataset. Each row shows the point cloud pair matching results at different distances.

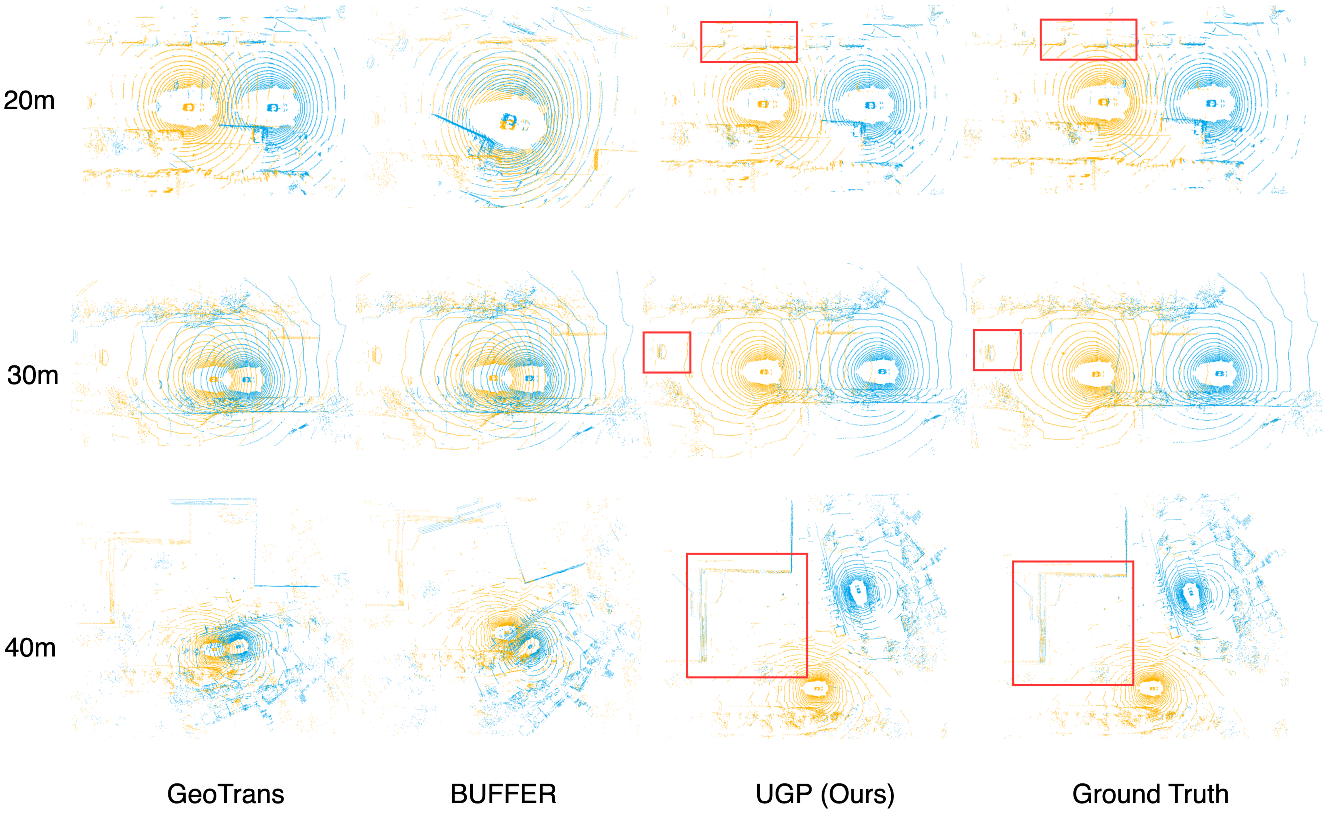


Figure 10. **Cross-distance generalization visualization** of GeoTrans [28], BUFFER [2], and UGP **on the nuScenes [4]** dataset. Each row presents the point cloud pair matching results at different distances.