RegFormer: An Efficient Projection-Aware Transformer Network for Large-Scale Point Cloud Registration

Jiuming Liu[1], Guangming Wang[1,4], Zhe Liu[2*], Chaokang Jiang[3], Marc Pollefeys[4,5], Hesheng Wang[1*]

[1]Department of Automation, Key Laboratory of System Control
and Information Processing of Ministry of Education, Shanghai Jiao Tong University
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3] China University of Mining and Technology [4] ETH Zürich [5] Microsoft

{liujiuming,wangguangming,liuzhesjtu,wanghesheng}@sjtu.edu.cn
ts20060079a31@cumt.edu.cn   marc.pollefeys@inf.ethz.ch

Abstract

Although point cloud registration has achieved remarkable advances in object-level and indoor scenes, large-scale registration methods are rarely explored. Challenges mainly arise from the huge point number, complex distribution, and outliers of outdoor LiDAR scans. In addition, most existing registration works generally adopt a two-stage paradigm: They first find correspondences by extracting discriminative local features and then leverage estimators (eg. RANSAC) to filter outliers, which are highly dependent on well-designed descriptors and post-processing choices. To address these problems, we propose an end-to-end transformer network (RegFormer) for large-scale point cloud alignment without any further post-processing. Specifically, a projection-aware hierarchical transformer is proposed to capture long-range dependencies and filter outliers by extracting point features globally. Our transformer has linear complexity, which guarantees high efficiency even for large-scale scenes. Furthermore, to effectively reduce mismatches, a bijective association transformer is designed for regressing the initial transformation. Extensive experiments on KITTI and NuScenes datasets demonstrate that our RegFormer achieves competitive performance in terms of both accuracy and efficiency. Codes are available at https://github.com/IRMVLab/RegFormer.

1. Introduction

Point cloud registration is a fundamental problem in 3D computer vision, which aims to estimate the rigid transformation between point cloud frames. It is widely applied in moblie robotics [22, 34], autonomous driving [47, 37], etc.
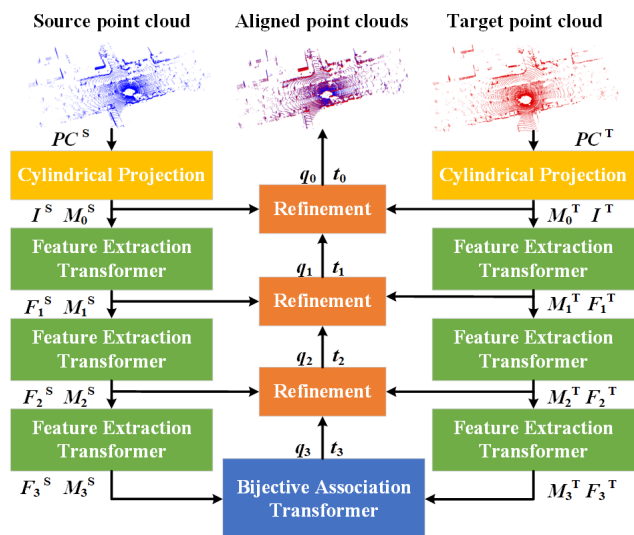

Figure 1. Overview architecture of RegFormer. The whole feature extraction and frame association sections are transformer-based. We project point cloud onto a 2D surface and feed its patches into transformer. A projection mask $M^T(M^S)$ is also proposed, which equips our transformer with the awareness of invalid positions.

Although learning-based methods show great potential in object-level or indoor registration tasks [42, 10, 1, 17], large-scale point cloud registration is less studied. Challenges are mainly three-fold: 1) Outdoor LiDAR scans may consist of hundreds of thousands of unstructured points, which are intrinsically sparse, irregular, and have a large spatial range. It is non-trivial to efficiently process all raw points in one inference [44]. 2) Outliers from dynamic objects and occlusion would degrade the registration accuracy as they introduce uncertain motions and inconsistency. 3) There are numerous mismatches when directly leveraging distance-based nearest neighbor matching methods (eg. kNN) to distant point cloud pairs [25].

---

*Corresponding Authors.   The first two authors contributed equally.

For the first challenge, previous registration works mostly voxelize input points [4, 25], and then establish putative correspondences by selecting keypoints and learning distinctive local descriptors [42, 10, 1]. However, quantization errors are inevitable in the voxelization [18]. Also, different selected keypoints may influence registration accuracy and downsampling challenges the repeatability [45]. In this paper, instead of searching keypoints, we directly process all LiDAR points by projecting them onto a cylindrical surface for the structured organization. Projected image-like structure facilitates the window partition in transformer, realizing linear computational costs. This enables our network to process almost 120000 points with high efficiency. To take advantage of 3D geometric features, each projected position is filled with raw point coordinates, inspired by [37]. Another concern is that projected pseudo images are full of invalid positions due to the original sparsity of point clouds. We handle this by designing a projection mask.

For the second challenge, the commonly used method is applying the robust estimator (RANSAC) [13, 4, 1] to filter outliers. However, RANSAC suffers from slow convergence [30] and is highly dependent on post-processing choices [44]. From a different view, we observe that global modeling capability is rather helpful to localize occluded objects and recognize dynamics as they introduce inconsistent global motion. Therefore, we propose a projection-aware transformer to extract point features globally. Notably, some recent works [30, 44] also try to design RANSAC-free registration networks. However, the combination of CNN and transformer in their feature extraction modules deteriorates the efficiency. The closest approach to ours is REGTR [44], which directly predicts clean correspondences with transformer. Nonetheless, the quadratic complexity limits its ability for large-scale application.

In addition, a Bijective Association Transformer (BAT) is designed to tackle the third challenge. HRegNet [25] already has awareness that nearest-neighbor matching can lead to considerable mismatches due to possible errors in descriptors. However, their kNN cluster is still distance-based, which can not generalize well to low-overlap inputs. To address this problem, two effective components are designed in BAT for reducing mismatches. The cross-attention mechanism is utilized first for preliminary location information exchange. Intuitively, features of deeper layers are coarse but reliable as they gather more information with larger receptive fields. Thus, each point is correlated with all points (instead of selecting k points) in the other frame to gain reliable motion embeddings on the coarsest layer (all-to-all). The precise transformation will then be recovered by the iterative refinement on shallow layers.

Overall, our contributions are as follows:

- We propose a fully end-to-end network for large-scale point cloud registration. It does not need any keypoint

matching or post-processing, which is both keypoint-free and RANSAC-free. Our efficient model can process hundreds of thousands of points in real time.
- The global modeling capability of our RegFormer can filter outliers effectively. Furthermore, a Bijective Association Transformer (BAT) is designed to reduce mismatches by combining cross-attention with an all-to-all point correlation strategy on the coarsest layer.
- Experiment results on KITTI [16, 15] and NuScenes [5] datasets indicate that our RegFormer achieves state-of-the-art performance with 99.8% and 99.9% successful registration recall respectively.

2. Related Work

Deep point cloud registration. Existing deep point cloud registration networks can be divided into two categories according to whether they extract explicit correspondences. The first class attempts to establish point correspondences through keypoint detection [26, 14, 25] and learning powerful discriminative descriptors [42, 10, 4, 1, 12]. As a pioneering work, 3DMatch [46] extracts local volumetric patch features by a Siamese network. PPFNet [12] and its unsupervised version PPF-FoldNet [11] extract global context-aware descriptors using PointNet [28]. To enlarge the receptive field, FCGF [10] computes dense descriptors of whole input point clouds in a forward pass. Recent correspondence-based networks [9, 41, 3, 6, 7] commonly use it to generate putative correspondences.

The second class directly estimates transformation in an end-to-end manner [2, 39, 19, 43]. Point clouds are aligned with learned soft correspondences or without explicit correspondences. Among these, PointNetLK [2] is a landmark that extracts global descriptors and estimates transformation with Lucas-Kanade algorithm [27]. FMR [19] enforces the optimization of registration by minimizing feature-metric projection errors. However, these direct registration methods can not generalize well to large-scale scenes [17]. Our method falls into this category and is specially designed for large-scale registration.

Large-scale point cloud registration. Large-scale registration is less studied in previous works. DeepVCP [26] incorporates both the local similarity and global geometric constraints in an end-to-end manner. Although it is evaluated on outdoor benchmarks, its keypoint matching is still constrained in local space. With a larger keypoint detection range, HRegNet [25] introduces bilateral and neighborhood consensus into keypoint features and achieves state-of-the-art. Different from previous works [26, 25, 40] that mostly focus on local descriptors, we address the issue from a more global perspective thanks to the long-range dependencies capturing ability of transformer. Motivated by recent scene flow works [35, 36], our RegFormer has no need for searching keypoint and explicit point-to-point correspondences,
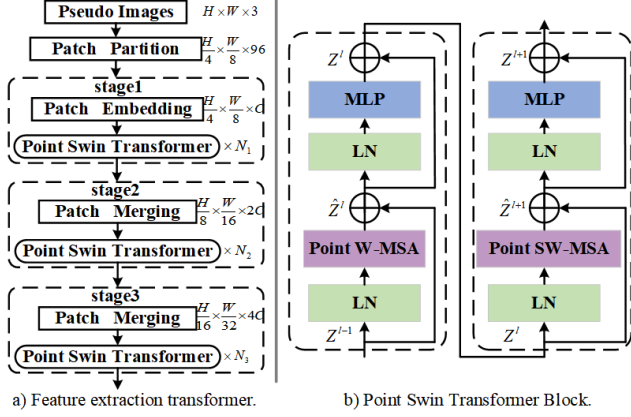
a) Feature extraction transformer. b) Point Swin Transformer Block.

Figure 2. The feature extraction module consists of three cascaded stages as constructed in a). b) indicates Point Swin Transformer block in stage l, which computes attention within windows (Point W-MSA), and then gathers contextual information by the spatial shift (Point SW-MSA).

which learns implicit cross-frame motion and directly outputs pose in a single pass.

Transformer in registration tasks. Most existing works [39, 14, 44, 45] only treat transformer as a frame association module. Among these, DCP [39] first utilizes a vanilla transformer to correlate downsampled features. RGM [14] proposes a framework based on deep graph matching, where transformer is employed to dynamically learn the soft edges of nodes. REGTR [44] outputs overlap scores and the location information through cross-attention, but it can not handle large-scale scenes. Our Reg-Former achieves linear complexity by revisiting attention within non-overlapping windows. Transformer is designed for not only frame association but also feature extraction. To the best of our knowledge, our RegFormer is the first pure transformer-based registration network.

3. RegFormer

3.1. Overall Architecture

The overall architecture of our proposed RegFormer is illustrated in Fig. 1. Given two point cloud frames: source point cloud $PC^S \in \mathbb{R}^{N \times 3}$ and target point cloud $PC^T \in \mathbb{R}^{M \times 3}$, the objective of registration is to align them via an estimated transformation. To orderly organize raw irregular points, we first project point clouds as pseudo images I^S and I^T in Section 3.2, and then feed them together with corresponding masks M_0^S and M_0^T into the hierarchical feature extraction module as in Fig. 2 a). Following prior works [23, 24], we treat each patch of size $4 \times 8 \times 3$ as a token, and then a feature embedding layer is utilized to project these patches to an arbitrary dimension denoted by C. The patch merging layer of each stage concatenates 2×2 neighbor patches. Then, concatenated features are reduced to half channels, and then fed into a projection-aware transformer
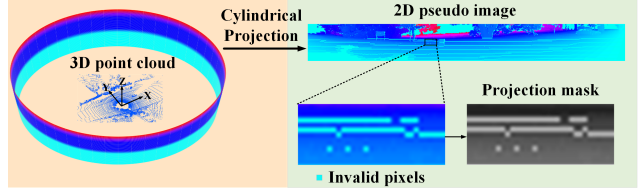

Figure 3. Cylindrical projection. We project 3D point clouds onto a 2D surface and fill each pixel with its raw x, y, z coordinates. A projection mask is also proposed to remove invalid positions.

in Section 3.3. For associating point clouds and reducing mismatches, a Bijective Association Transformer (BAT) in Section 3.4 is employed to generate initial motion embeddings. Finally, the quaternion $q_3 \in \mathbb{R}^4$ and translation vector $t_3 \in \mathbb{R}^3$ are estimated from motion embeddings, and then refined iteratively.

3.2. Cylindrical Projection

According to the original proximity relationship of raw points, point clouds are projected onto a cylindrical surface, following the line scanning characteristic of the LiDAR sensor. Each point has a corresponding 2D pixel position on projected pseudo images as:

$$u = arctan2(y/x)/\Delta\theta, \tag{1}$$

$$v = arcsin(z/\sqrt{x^2 + y^2 + z^2})/\Delta\phi, \tag{2}$$

where x, y, z represent the raw 3D coordinates of point cloud and u, v are corresponding 2D pixel positions. $\Delta\theta, \Delta\phi$ are horizontal and vertical resolutions of the Li-DAR sensor. To make the best use of geometric information of raw 3D points, we fill each pixel position with its raw x, y, z coordinates. Pseudo images of size $H \times W \times 3$ in Fig. 3 will be input to the feature extraction transformer.

3.3. Point Swin Transformer

Compared with images, the scale of outdoor LiDAR points is surprisingly larger, and thus they require a much larger number of tokens for representation. Vanilla transformer with quadratic complexity is not suitable, as it will lead to huge computation costs. Inspired by Swin Transformer [23], we introduce window attention into 3D point transformer for linear complexity. Thanks to the global understanding ability of transformer, our network can effectively learn to identify dynamic motions and the location of occluded objects in the other frame. To simplify the formulation, we only expand the description for the source point cloud in this section, and the same goes for the target one.

Projection masks. It is non-trivial to extend the 2D window-based attention mechanism to the pseudo images generated from 3D points. Point cloud, especially in outdoor scenes, is extremely sparse. Thus, projected pseudo images are filled with invalid blank pixels. The registration
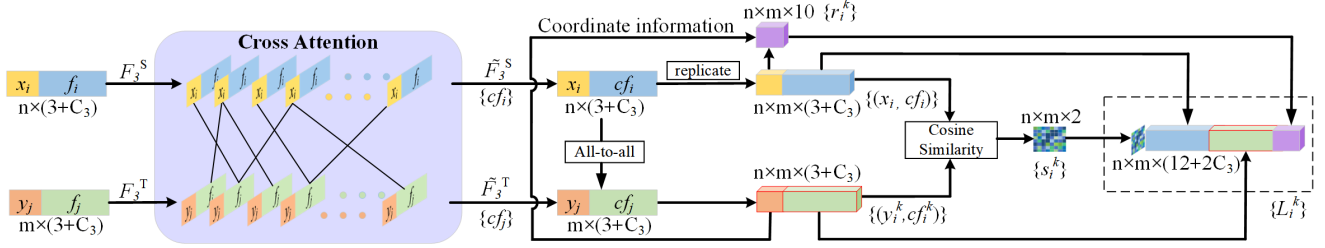
Figure 4. Bijective Association Transformer. The cross-attention mechanism is leveraged for preliminary information exchange between two frames. Then, geometric characteristics of conditioned features $\tilde{F}_3^S$, $\tilde{F}_3^T$ are fully considered to generate the initial motion embeddings.

accuracy will be affected if they are fed identically into attention. Also, the attention calculation itself of these pixels is meaningless as they have no corresponding raw points. Inspired by [8], a projection-aware binary mask M_l^S is proposed here, which represents whether each pixel is invalid in Fig. 3:

$$M_l^S = \begin{cases} -\infty, & x = 0, y = 0, z = 0 \,, \\ 0, & otherwise \,, \end{cases} \quad (3)$$

where x, y, z are point coordinates filled into pseudo images. The projection mask M_l^S of size $H \times W \times 1$ is pixel-corresponding to the pseudo image and together downsampled (stride-based) in each stage as Fig. 1. l denotes the stage number. We assign zero to valid pixels and a big negative number to invalid ones where attention should not be calculated. In this way, invalid pixels would then be filtered through the softmax operation afterward in attention blocks.

Point W-MSA and Point SW-MSA. For stage l, point feature Z^{l-1} ($H_l \times W_l \times C_l$) and its corresponding mask M_l^S ($H_l \times W_l \times 1$) are fed into Point Window-based Multi-Head Self Attention (Point W-MSA) as:

$$W\text{-}MSA(Z^{l-1}) = (Head_1 \oplus \cdots \oplus Head_H)W^O, \quad (4)$$

$$Head_h = Attention(Q^h, K^h, V^h)$$
$$= softmax(\frac{Q^h K^h}{\sqrt{d_{head}}} + M_l^S + Bias)V^h, \quad (5)$$

where $Head_h$ represents the output of h-th head. M_l^S is the projection mask. $\oplus$ denotes the concatenation of two vectors. $Bias$ is the relative position encoding [31]. $Q^h = Z^{l-1}W_h^Q$, $K^h = Z^{l-1}W_h^K$, $V^h = Z^{l-1}W_h^V$, in which $W_h^Q \in \mathbb{R}^{C_l \times C_{head}}$, $W_h^K \in \mathbb{R}^{C_l \times C_{head}}$, $W_h^V \in \mathbb{R}^{C_l \times C_{head}}$, $W^O \in \mathbb{R}^{HC_{head} \times C_l}$ are learned projections. The above process is repeated again in the following Point Shift Window-based Multi-head Self Attention (Point SW-MSA). The only difference is that features are first spatially shifted [23] in Point SW-MSA for increasing information interaction among windows.

Point Swin Transformer blocks. Overall, one complete transformer stage in Fig. 2 b) can be described as:

$$\hat{Z}^l = PW\text{-}MSA(LN(Z^{l-1})) + Z^{l-1}$$
$$Z^l = MLP(LN(\hat{Z}^l)) + \hat{Z}^l$$
$$\hat{Z}^{l+1} = PSW\text{-}MSA(LN(Z^l)) + Z^l$$
$$Z^{l+1} = MLP(LN(\hat{Z}^{l+1})) + \hat{Z}^{l+1}, \quad (6)$$

where P(S)W-MSA represents Point (Shift) Window Multi-head Self Attention. Z^{l+1} is the output feature of stage l.

3.4. Bijective Association Transformer

After global features are hierarchically extracted by our Point Swin Transformer, the key issue is how to match source and target point clouds through their downsampled features. The most common method is to search for nearest neighbors (NN). However, this distance-dependent strategy is ineffective enough for large-scale registration, as two corresponding points may be too far away, which leads to numerous mismatches [25]. To solve this problem, we propose a Bijective Association Transformer block (BAT) in Fig. 4, which first learns rough but generally correct location information with cross-attention. Then, an all-to-all point gathering strategy guarantees reliable location output and further reduces mismatches on the coarsest layer.

Rough association. As depicted in Fig. 4, downsampled source and target point features of stage 3 are first fed into a cross-attention layer with linear complexity, roughly associating with each other. Cross attention can introduce certain similarities of two point cloud frames by calculating attention weights and updating features with the awareness of point location in the other frame [44]. Concretely, source and target point coordinates and their features are first resized as $X_3^S \in \mathbb{R}^{n \times 3}$, $Y_3^T \in \mathbb{R}^{m \times 3}$ and $F_3^S \in \mathbb{R}^{n \times C_3}$, $F_3^T \in \mathbb{R}^{m \times C_3}$, which are inputs of the cross-attention block. The output conditioned features $\tilde{F}_3^S$ for source point cloud can be written as:

$$\tilde{F}_3^S = Attention(F_3^S W^Q, F_3^T W^K, F_3^T W^V), \quad (7)$$

where W^Q, W^K, W^V are respective projected functions. $Attention$ is similar to Section 3.3. When the source point cloud serves as $query$, the target point cloud would be projected as key and $value$, and vice versa.

All-to-all point gathering. The coarsest layer obviously gathers more information and a larger receptive field,

which is reliable to match two frames. Thus, on the bottom layer of our RegFormer, each point in $\tilde{F}_3^S$ is associated with all points in $\tilde{F}_3^T$, rather than select k nearest neighbor points (kNN), to generate reliable motion embeddings. Specifically, each point in $PC^S = \{(x_i, f_i)|x_i \in X^S, cf_i \in \tilde{F}_3^S, i = 1, \cdots, n\}$ correlates with all m points in $PC^T = \{(y_j, f_j)|y_j \in Y^T, cf_j \in \tilde{F}_3^T, j = 1, \cdots, m\}$, forming an association cluster $\{(y_i^k, cf_i^k)|k = 1, \cdots, m\}$. Then, the relative 3D Euclidean space information $\{r_i^k\}$ is calculated as:

$$r_i^k = x_i \oplus y_i^k \oplus (x_i - y_i^k) \oplus \|x_i - y_i^k\|_2, \quad (8)$$

where $\|\cdot\|_2$ indicates the L_2 Norm.

The cosine similarity of grouped features is also introduced as:

$$s_i^k = \frac{< cf_i, cf_i^k >}{\|cf_i\|_2 \|cf_i^k\|_2}, \quad (9)$$

where $<,>$ denotes the inner product. This step will output a $n \times m \times 2$ similarity feature s_i^k, where the neighbor similarity in [25] is also considered here.

Then, we concatenate the above space and similarity embeddings and utilize a 3-layer shared MLP on them:

$$L_i^k = MLP(f_i \oplus cf_i^k \oplus r_i^k \oplus s_i^k). \quad (10)$$

Finally, the initial flow embedding can be represented by the attentive encoding of concatenated features as:

$$fe_i = \sum_{k=1}^{m} L_i^k \odot \underset{k=1,\cdots,m}{softmax}(L_i^k), \quad (11)$$

where the output motion embedding fe_i is a weighted sum of L_i^k by the softmax function. $\odot$ represents dot product.

3.5. Estimation of the Rigid Transformation

The transformation is estimated from initial motion embeddings $FE = \{fe_i, i = 1, \cdots, n\}$ together with downsampled source point features F_3^S in layer 3 as:

$$W = softmax(MLP(FE \oplus F_3^S)), \quad (12)$$

where $W = \{w_i|w_i \in \mathbb{R}^{C_3}\}$ are attention weights. Then, the quaternion $q_3 \in \mathbb{R}^4$ and translation vector $t_3 \in \mathbb{R}^3$ can be generated separately from weighting and sum operations followed by a fully connected layer:

$$q_3 = \frac{FC_1(\sum_{i=1}^{n} fe_i \odot w_i)}{|FC_1(\sum_{i=1}^{n} fe_i \odot w_i)|}, \quad (13)$$

$$t_3 = FC_2(\sum_{i=1}^{n} fe_i \odot w_i), \quad (14)$$

where FC_1 and FC_2 denote two fully connected layers.

Nonetheless, the initially estimated transformation is not precise enough due to the sparsity of the coarsest layer.

Thus, we iteratively refine it on upper layers with PWC structure [32, 38] to generate residual transformation Δq^l and Δt^l. Refinement in the l-th layer can be indicated as:

$$q^l = \Delta q^l q^{l+1}, \quad (15)$$

$$[0, t^l] = \Delta q^l [0, t^{l+1}](\Delta q^l)^{-1} + [0, \Delta t^l]. \quad (16)$$

3.6. Loss Function

Our network outputs transformation parameters from four layers and adopts a multi-scale supervised approach: $L = \alpha^l \mathcal{L}^l$. α^l indicates weights of layer l. $\mathcal{L}^l$ denotes the loss function of the l-th layer, which is calculated as:

$$\mathcal{L}^l = \mathcal{L}_{trans}^l exp(-k_t) + k_t + \mathcal{L}_{rot}^l exp(-k_r) + k_r, \quad (17)$$

where k_t and k_r are two learnable parameters, which can uniform the difference in the unit and scale between quaternion and translation vectors [21]. $\mathcal{L}_{trans}^l$ and $\mathcal{L}_{rot}^l$ can be calculated as:

$$\mathcal{L}_{trans}^l = \|t^l - \hat{t}^l\|, \quad (18)$$

$$\mathcal{L}_{rot}^l = \|\frac{q^l}{\|q^l\|} - \hat{q}^l\|_2, \quad (19)$$

where $\|\cdot\|$ indicates the L_1 Norm. q^l, t^l and $\hat{q}^l, \hat{t}^l$ are estimated and ground truth transformations respectively.

4. Experiment

We evaluate our RegFormer on two large-scale point cloud datasets, namely KITTI [16, 15] and NuScenes [5]. Moreover, ablation studies are conducted for each designed component of our network to demonstrate their effectiveness. Extensive experiments demonstrate that our methods can achieve state-of-the-art registration accuracy and also guarantee high efficiency.

4.1. Experiment Settings

Implement Details. In the data processing, we directly input all LiDAR points without downsampling. Projected pseudo images are set in line with the range of LiDAR sensor as 64 (H)×1792 (W) for KITTI and 32 (H)×1792 (W) for NuScenes. Window size and shift size are set as 4 and 2 separately. Experiments are conducted on a single NVIDIA Titan RTX GPU with PyTorch 1.10.1. The Adam optimizer is adopted with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The initial learning rate is 0.001 and exponentially decays every 200000 steps until 0.00001. The batch size is set as 8. The hyperparameter α^l in the loss function is set to 1.6, 0.8, 0.4, and 0.2 for four layers. Initial values of learnable parameters k_t and k_r are set as 0.0 and -2.5 respectively. More experiment details are presented in Appendix.

Evaluation metrics. We follow protocols of DGR [9] to evaluate our RegFormer with three metrics: (1) Relative Translation Error (RTE). (2) Relative Rotation Error (RRE). (3) Registration Recall (RR). RR is defined when RRE and RTE are within a certain threshold.

Method	RTE(m)		RRE(°)		RR(%)	Time(ms)/Points	NT(ms)
	AVG	STD	AVG	STD			
FGR [48]	0.93	0.59	0.96	0.81	39.4%	506.1/11445	44.22
RANSAC [13]	0.13	0.07	0.54	0.40	91.9%	549.6/16384	33.54
DCP [39]	1.03	0.51	2.07	1.19	47.3%	46.4/1024	45.31
IDAM [20]	0.66	0.48	1.06	0.94	70.9%	33.4/4096	8.15
FMR [19]	0.66	0.42	1.49	0.85	90.6%	85.5/12000	7.13
DGR [9]	0.32	0.32	0.37	0.30	98.7%	1496.6/16384	91.35
HRegNet [25]	0.12	0.13	0.29	0.25	99.7%	106.2/16384	6.48
Ours	**0.08**	**0.11**	**0.23**	**0.21**	**99.8%**	**98.3/120000**	**0.82**

Table 1. Comparison with state-of-the-art. The best performance is highlighted in bold. Registration Recall (RR) is defined as the success ratio where RRE< 5° and RTE< 2m. 'NT' means normalized time per thousand points.
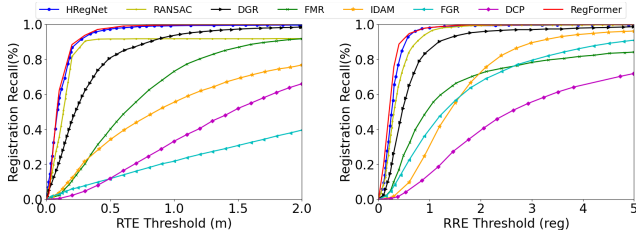

Figure 5. Registration recall with different RRE and RTE thresholds on the KITTI dataset.

4.2. KITTI Benchmark

KITTI odometry dataset is composed of 11 sequences (00-10) with ground truth poses. Following the settings in [10, 9], we use 00-05 for training, 06-07 for validation, and 08-10 for testing. Also, the ground truth poses of KITTI are refined with ICP [10, 4].

Comparison with state-of-the-art. Following [25], the current frame and the 10th frame after it are used to form input point pairs. We choose both traditional and learning-based registration methods for comparison. For classical methods, our network is superior to FGR [48] by a large margin in both accuracy and efficiency. Compared with RANSAC [13], our RegFormer has a 7.9% higher RR. Also, RANSAC suffers from much lower efficiency (five times more total time than ours) due to the slow convergence. With respect to learning-based methods, RegFormer is compared with a series of state-of-the-art. DCP [39], IDAM [20], and FMR [19] are all feature-based registration networks that extract local descriptors. DGR [9] achieves competitive performance in indoor scenes with global features. As in Table 1, our RegFormer has much lower RRE and RTE, higher RR than all the above learning-based methods without designing discriminative descriptors. HRegNet [25] is recent CNN-based SOTA for outdoor large-scale scenes. Our RegFormer is more accurate in terms of all metrics and has a 7.2% efficiency improvement than theirs. As for efficiency, both total time and normalized time are given. Normalized time is calculated by the processing speed per thousand points. As illustrated in Table 1, our RegFormer can process large-scale points with the highest

Method	Backbone	RTE(cm)	RRE(°)	RR(%)	Time(s)
3DFeat-Net [42]	CNN	25.9	0.25	96.0%	3.4
FCGF [10]	CNN	9.5	0.30	96.6%	3.4
D3Feat [4]	CNN	7.2	0.30	99.8%	3.1
Predator [17]	CNN	**6.8**	0.27	99.8%	5.2
CoFiNet [45]	CNN	8.5	0.41	99.8%	1.9
SpinNet [1]	CNN	9.9	0.47	99.1%	60.6
GeoTransformer [30]	Transformer	7.4	0.27	99.8%	1.6
Ours	Transformer	8.4	**0.24**	**99.8%**	**0.1**

Table 2. Comparison with RANSAC-based networks. The best performance is highlighted in bold. RR is defined as the success ratio where RRE< 5° and RTE< 2m.

average efficiency (0.82ms). Registration recalls with different RRE and RTE thresholds are also displayed in Fig. 5, which proves that our RegFormer is extremely robust to various threshold settings.

Comparison with RANSAC-based models. RANSAC is a commonly employed estimator for filtering outliers. With no need for RANSAC, our RegFormer leverages the attention mechanism for improving the resilience to outliers by learning global features. Its effectiveness is demonstrated by the comparison with RANSAC-based methods in Table 2. We follow settings in [30] using input point pairs at least 10m away and setting the RTE threshold as 2m. Also, all methods are divided into two categories in terms of different backbones: CNN and Transformer. Our network is on par with all SOTA CNN-based works including 3DFeatNet [42], FCGF [10], D3Feat [4], CoFiNet [45], Predator [17], and SpinNet [1]. Notably, although Predator has 1.6 cm lower RTE than ours due to well-designed local descriptors, it has 11.1% larger RRE and 52× more runtime compared with ours. In terms of transformer-based networks, GeoTransformer [30] introduces geometric features into transformer and has a marginally smaller RTE, but it has a higher RRE and obvious efficiency decline. Our efficient RegFormer has a 16× speed-up compared with theirs.

4.3. NuScenes Benchmark

We further evaluate our RegFormer on NuScenes. It consists of 1000 scenes, including 850 scenes for training and validation, and 150 scenes for testing. Following [25], we use 700 scenes for training, 150 scenes for validation, and 10th frame away point pairs.

Comparison with state-of-the-art. As illustrated in Table 3, our registration accuracy outperforms all classical works and most learning-based ones. Our RegFormer has a 39% RR improvement compared with RANSAC with only 30% of their runtime. For learning-based methods, our model is superior to DCP [39], IDAM [20], and FMR [19] by a large margin. Compared with HRegNet [25], our RegFormer has the same 99.9% recall. Because the feature extraction section is finely pre-trained, it has 0.02m lower RTE than ours, but more than two times RRE (0.45°) instead. Moreover, their efficiency is lower than ours.

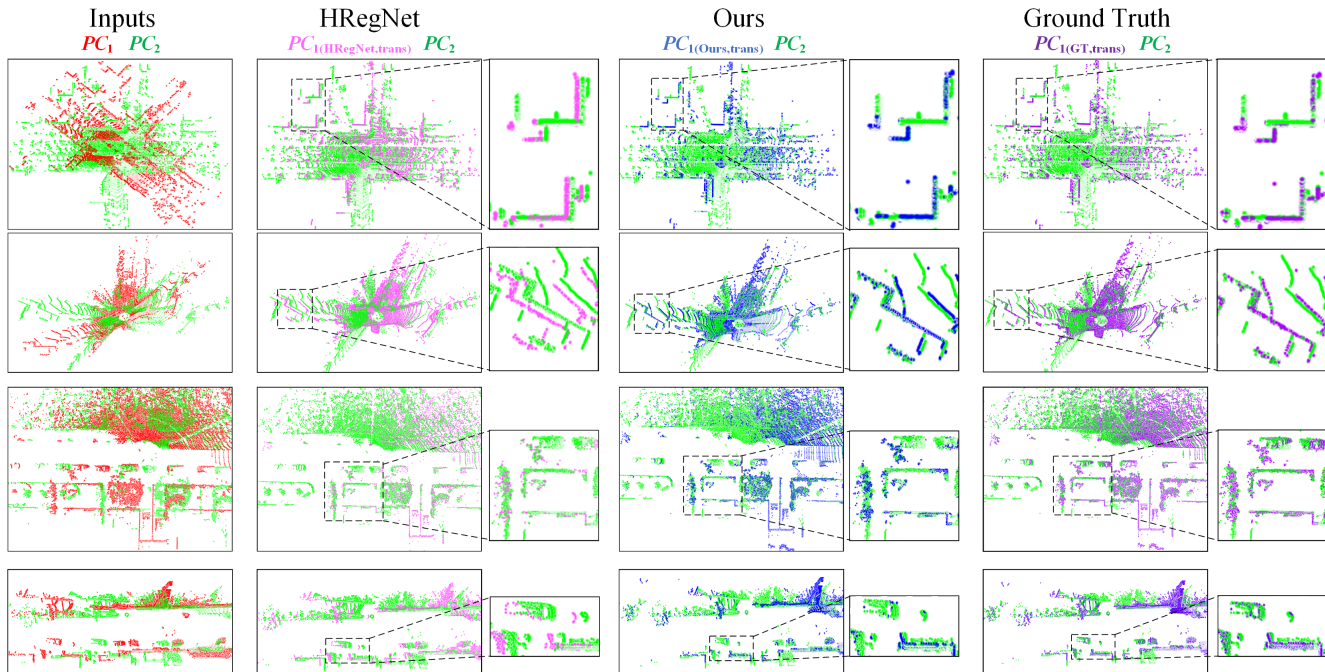| Inputs PC_1 PC_2 | HRegNet $PC_{1(HRegNet,trans)}$ PC_2 | Ours $PC_{1(Ours,trans)}$ PC_2 | Ground Truth $PC_{1(GT,trans)}$ PC_2 |

Figure 6. Low-overlap registration results. Point clouds colored red and green indicate the input source and target point clouds. Transformed source points by the estimated pose of ours and HRegNet are colored blue and pink respectively. The ground truth is colored purple. Our RegFormer can align low-overlap input points even with large translations (upper two rows) or rotations (lower two rows).

Method	RTE(m)	RRE(°)	Recall(%)	Time/Points	NT(ms)
FGR [48]	0.71	1.01	32.2%	284.6/11445	24.87
RANSAC [13]	0.21	0.74	60.9%	268.2/8192	32.74
DCP [39]	1.09	2.07	58.6%	45.5/1024	44.43
IDAM [20]	0.47	0.79	88.0%	32.6/4096	7.96
FMR [19]	0.60	1.61	92.1%	61.1/12000	5.09
DGR [9]	0.21	0.48	98.4%	523.0/8192	63.84
HRegNet [25]	**0.18**	0.45	**99.9%**	87.3/8192	10.66
Ours	0.20	**0.22**	**99.9%**	**85.6/50000**	**1.71**

Table 3. Quantitative results on NuScenes. The best performance is in bold. 'NT' means normalized time per thousand points.
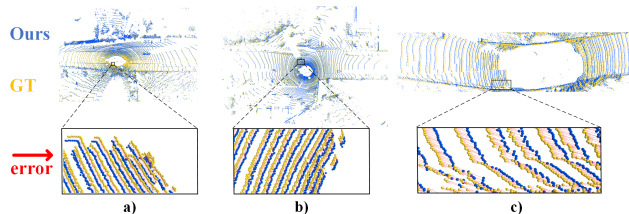

Figure 7. Visualization of registration errors. Point clouds colored yellow and blue indicate transformed source points by the ground truth (GT) and our estimated pose. Registration errors are visualized by a red vector pointing from estimated points to the GT.

4.4. Qualitative Visualization

Low-overlap Registration. Fig. 6 selects four challenging samples of registration results on the KITTI dataset. Our RegFormer can effectively align source and target point clouds even though they originally have large translations or rotations with low overlap. Also, our RegFormer has higher registration accuracy compared with HRegNet [25].

Visualization of the registration errors. To further study the source of errors, error vectors are visualized in Fig. 7. Interestingly, our registration errors are also influenced by the surroundings due to the data-driven characteristics. When features on both sides of the vehicle are sufficient as Fig. 7 a) and b), errors are relatively distributed evenly. It can be attributed to the structured buildings around, which offer solid positioning information. However, if there are scarce reference objects besides the car or surrounding features are monotonous as in Fig. 7 c), errors mainly come from the front and rear directions.

4.5. Ablation Study

In this section, extensive ablation studies are conducted for each designed element.

Hierarchical architecture. We separately output estimated transformation parameters from coarser layers and re-evaluate the metrics. As displayed in Table 4, rotation and translation are generated from layer 3 (a), layer 2 (b), and layer 1 (c). It is obvious that registration errors get smaller as the transformation is iteratively refined.

Projection masks. The projection mask in our transformer is removed to evaluate its effectiveness. As in Table 5 (a), the whole registration accuracy drops dramatically since numerous invalid pixels are also taken into account.

Global modeling capability. Different from most previ-

Model	RTE(m)	RRE(°)	Recall(%)
(a) Transformation from layer 3	0.96 ± 0.42	1.49 ± 1.03	64.2%
(b) Transformation from layer 2	0.75 ± 0.44	1.31 ± 0.85	79.9%
(c) Transformation from layer 1	0.38 ± 0.30	0.88 ± 0.82	96.8%
Ours (from layer 0)	**0.08 ± 0.11**	**0.23 ± 0.21**	**99.8%**

Table 4. Ablation studies of the hierarchical architecture.

Model	RTE(m)	RRE(°)	Recall(%)
(a) w/o projection mask	0.22 ± 0.18	0.52 ± 0.53	98.1%
(b) replace transformer with 2D CNN	0.57 ± 0.42	0.89 ± 0.81	82.2%
(c) replace transformer with PointNet++ [29]	0.24 ± 0.25	0.57 ± 0.62	92.4%
(d) w/o cross attention in BAT	0.19 ± 0.17	0.48 ± 0.44	98.7%
(e) w/o all-to-all points gathering in BAT	0.88 ± 0.46	1.88 ± 1.02	63.3%
(f) replace BAT with cost volume in [38]	0.75 ± 0.49	1.20 ± 0.90	60.3%
(g) replace BAT with cost volume in [33]	0.18 ± 0.22	0.33 ± 0.46	93.4%
Ours (Full)	**0.08 ± 0.11**	**0.23 ± 0.21**	**99.8%**

Table 5. Ablation studies of components in transformer module.

ous works, our RegFormer focuses on more global features with transformer. The global modeling ability enables our network to sufficiently capture dynamics and recover the occluded objects. To verify this, we replace the feature extraction transformer with CNNs, keeping other components unchanged. As indicated in Table 5 (b), 2D CNN is used to extract local features with the same downsampling scale. PointNet++ [29] is also utilized to replace our transformer in Table 5 (c). The results show both local features extracted by 2D CNN and Pointnet++ have larger registration errors since their receptive fields are constrained.

Bijective Association Transformer (BAT). In this paper, cross-attention is leveraged to exchange information between frames in advance. Here, we remove the cross-attention in BAT to quantitatively test the effectiveness. Table 5 (d) shows that registration errors become double without cross-attention module. Also, the all-to-all point grouping strategy is extremely crucial to find reliable points and reduce mismatches as in Table 5 (e). Furthermore, we conduct experiments by replacing BAT with the cost volume mechanism [38, 33], which is commonly used for consecutive frame association. From the results in Table 5 (f) (g), we can witness at least 0.1m larger RTE and 6.4% RR drop.

5. Discussion

Here, we discuss why our RegFormer has such excellent accuracy. The competitive performance can be basically attributed to the outlier elimination capability and mismatch rejection strategy. We will elaborate on these two mechanisms in detail respectively.

Global modeling ability to filter outliers. Transformer can learn patch similarity globally while dynamics and occlusion have inconsistent global motion. So, our transformer-based pipeline can effectively recognize and eliminate interference from these objects by paying less attention to these patches as in Fig. 8. In this case, our Reg-
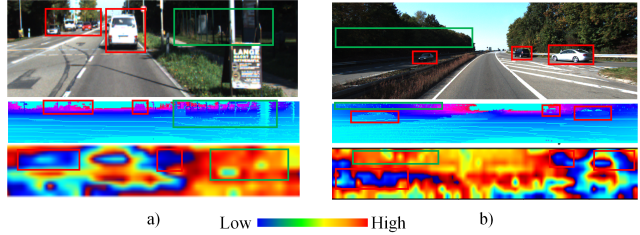

Figure 8. Visualization of attention weights. We give two samples here, where the first two rows respectively represent corresponding pictures and projected point clouds. Attention weights are visualized in the last row. Dynamic objects (red box) have lower attention weights, and static objects (green box) have higher weights.
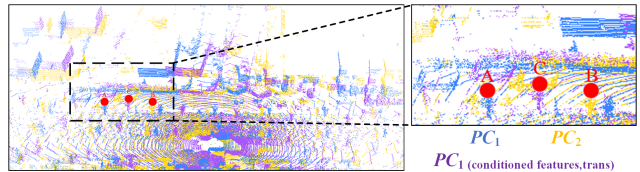

Figure 9. Visualization of the cross attention mechanism in BAT. Points A and B are corresponding points respectively in the source and target frame. Point C is the transformed position of A by conditioned features after cross-attention block.

Former can maintain high registration accuracy without robust estimators like RANSAC.

Cross-attention mechanism for reducing mismatches. In our Bijective Association Transformer module, a cross-attention block is first applied to exchange information and embed motion between two frames. Here, we remove the rest parts of BAT, leveraging only conditioned features from the cross-attention block to generate a directionally correct but not precise transformation. Then, it is used to transform input point clouds as in Fig. 9 (purple). For each point in the source point cloud (blue), its corresponding point in the target one (yellow) is originally almost 10m away. Cross-attention can effectively shorten this distance between two frames by learning preliminary motion embeddings.

6. Conclusion

In this paper, we proposed a transformer-based large-scale registration network. Global features are extracted by transformer to filter outliers. To cope with the irregularity and sparsity of raw point clouds, we leverage cylindrical projection to organize them orderly and present a projection mask to remove invalid pixels. Furthermore, a bijective association transformer, including cross-attention-based preliminary information exchange and all-to-all point gathering, is designed for reducing mismatches. The whole model is RANSAC-free, high-accuracy, and extremely efficient.

References

[1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11753–11762, 2021.

[2] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7163–7172, 2019.

[3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15859–15869, 2021.

[4] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6359–6367, 2020.

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[6] Wen Chen, Haoang Li, Qiang Nie, and Yun-Hui Liu. Deterministic point cloud registration via novel transformation decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6348–6356, 2022.

[7] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13221–13231, 2022.

[8] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021.

[9] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2514–2523, 2020.

[10] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019.

[11] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018.

[12] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 195–205, 2018.

[13] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[14] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8893–8902, 2021.

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[17] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4267–4276, 2021.

[18] Tianxin Huang, Jiangning Zhang, Jun Chen, Zhonggan Ding, Ying Tai, Zhenyu Zhang, Chengjie Wang, and Yong Liu. 3qnet: 3d point cloud geometry quantization compression network. *ACM Transactions on Graphics (TOG)*, 41(6):1–13, 2022.

[19] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11366–11374, 2020.

[20] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *European conference on computer vision*, pages 378–394. Springer, 2020.

[21] Qing Li, Shaoyang Chen, Cheng Wang, Xin Li, Chenglu Wen, Ming Cheng, and Jonathan Li. Lo-net: Deep real-time lidar odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8473–8482, 2019.

[22] Jiuming Liu, Guangming Wang, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Translo: A window-based masked point transformer framework for large-scale lidar odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1683–1691, 2023.

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[24] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[25] Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, and Rongqi Gu. Hregnet: A hierarchical net-

work for large-scale outdoor lidar point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16014–16023, 2021.

[26] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–21, 2019.

[27] Bruce D Lucas, Takeo Kanade, et al. *An iterative image registration technique with an application to stereo vision*, volume 81. Vancouver, 1981.

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[30] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11143–11152, 2022.

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[32] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[33] Guangming Wang, Yunzhe Hu, Zhe Liu, Yiyang Zhou, Masayoshi Tomizuka, Wei Zhan, and Hesheng Wang. What matters for 3d scene flow network. In *European Conference on Computer Vision*, pages 38–55. Springer, 2022.

[34] Guangming Wang, Yunzhe Hu, Xinrui Wu, and Hesheng Wang. Residual 3-d scene flow learning with context-aware feature extraction. *IEEE Transactions on Instrumentation and Measurement*, 71:1–9, 2022.

[35] Guangming Wang, Chaokang Jiang, Zehang Shen, Yanzi Miao, and Hesheng Wang. Sfgan: Unsupervised generative adversarial learning of 3d scene flow from the 3d scene self. *Advanced Intelligent Systems*, 4(4):2100197, 2022.

[36] Guangming Wang, Xiaoyu Tian, Ruiqi Ding, and Hesheng Wang. Unsupervised learning of 3d scene flow from monocular camera. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4325–4331. IEEE, 2021.

[37] Guangming Wang, Xinrui Wu, Shuyang Jiang, Zhe Liu, and Hesheng Wang. Efficient 3d deep lidar odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5749–5765, 2022.

[38] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Pwclo-net: Deep lidar odometry in 3d point clouds using

hierarchical embedding mask optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15910–15919, 2021.

[39] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019.

[40] Louis Wiesmann, Tiziano Guadagnino, Ignacio Vizzo, Giorgio Grisetti, Jens Behley, and Cyrill Stachniss. Dcpcr: Deep compressed point cloud registration in large-scale outdoor environments. *IEEE Robotics and Automation Letters*, 7(3):6327–6334, 2022.

[41] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020.

[42] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *Proceedings of the European conference on computer vision (ECCV)*, pages 607–623, 2018.

[43] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020.

[44] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6677–6686, 2022.

[45] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Advances in Neural Information Processing Systems*, 34:23872–23884, 2021.

[46] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017.

[47] Yuchao Zheng, Yujie Li, Shuo Yang, and Huimin Lu. Global-pbnet: A novel point cloud registration for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[48] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European conference on computer vision*, pages 766–782. Springer, 2016.