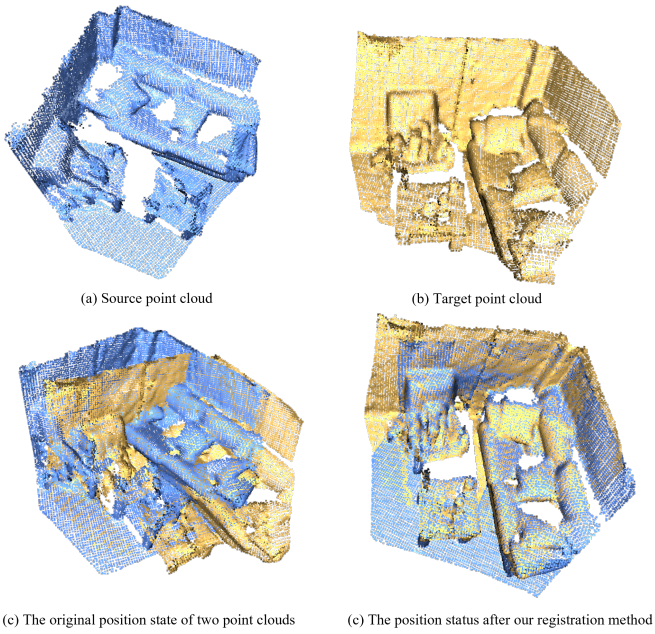
MS-Transformer: Masked and Sparse Transformer for Point Cloud Registration

Qingyuan Jia, Guiyang Luo, Quan Yuan, Jinglin Li, Congzhang Shao and Ziyue Chen
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
{qyjia, luoguiyang, yuanquan, jlli, shaocongzhang, bupt_czy}@bupt.edu.cn

***Abstract*—In this paper, we propose a masked and sparse transformer to address the problem of point cloud registration with low overlap. The mask mechanism reduces the overall data, increasing the corresponding point ratio in the overlap region, while also reducing the computational cost to accelerate the algorithm's execution speed. Moreover, we combine spatial position encoding and sparse self-attention to establish relationships within the source point cloud, as well as the relationships and attention scores between the source and target point clouds. This approach is specifically designed for the task of point cloud registration. Finally, we search for the maximum overlap area by matching the spatial consistency between points and calculate the 3D transformation matrix to complete the registration process. Our method achieves an improvement in the inlier ratio and performs well on the 3DMatch and 3DLoMatch datasets, demonstrating high registration efficiency.**

***Index Terms*—Point cloud registration, Self-attention, Mask mechanism**



(a) Source point cloud (b) Target point cloud

(c) The original position state of two point clouds (c) The position status after our registration method

Fig. 1. Figure (a) and (b) show the source and target point clouds, respectively. Figure (c) presents the initial position overlay of the two point clouds, while Figure (d) illustrates the pose transformation after registration.

I. Introduction

Point cloud registration involves estimating transformations between point clouds. By calculating the coordinate transformations, the process unifies the point cloud data from different viewpoints through rigid transformations such as rotation and translation to a specified coordinate system. Point cloud registration finds applications in various fields, including 3D reconstruction, parameter evaluation, localization, pose estimation, and autonomous driving [1], [2]. Autonomous driving relies on various sensors, with LiDAR [3] and RADAR [4] as important sensors to generate point cloud data. Combined with multi-view images captured by the camera [5], point cloud registration can lay an accurate foundation for subsequent segmentation, tracking [6], positioning [7], and target detection [8] tasks. Recently, the progress in point cloud registration has mainly been based on machine learning and correspondence-based methods [9]–[12]. Learning-based methods mostly extract the correspondence between two input point clouds by training a neural network and then calculating the pose transformation using a robust estimator such as RANSAC [13]. Correspondence-based methods rely mostly on keypoint detection, but this approach performs poorly in situations with low overlap because it is challenging to find repeatable keypoints in two point clouds. Learning-based methods have performed well in datasets with high overlap, and various state-of-the-art methods have achieved over 95% registration recall rate on 3DMatch [14] and KITTI [15] datasets. However, the accuracy significantly decreases in situations with low overlap, where the best-performing algorithm achieves only a 74% registration recall rate on the 3DLoMatch dataset. We have identified two factors contributing to this issue. Firstly, the presence of irrelevant data in the non-overlapping areas affects the registration process. Secondly, the number of matching points and corresponding features in the overlapping area is also reduced.

VIT [16] has successfully applied the transformer in image processing by dividing the image into small blocks and training them in the network. Similarly, in point cloud registration, after subsampling, superpoints which are groups of nearby points, can be used to form small patches. Sparse and loose superpoint matching reduces the strict point matching to overlapping blocks, thereby relaxing the requirement for

repeatability. Compared with distance-based point matching, block overlap matching provides more reliable and informative correspondence constraints. However, superpoint matching requires capturing more global context features. The Transformer can encode contextual information in point cloud registration [17]. Nonetheless, traditional Transformers overlook the geometric structure of point clouds [18], resulting in reduced discriminability in learned features and a substantial number of outliers in the matching process. Some existing methods [17], [19] incorporate geometric position coding or combine with DGCNN [20] to introduce geometric features into the learning process, but they do not fully consider the effect of non-overlapping areas on registration. Transformers have a large computational cost and are inefficient when dealing with large point cloud inputs, and most superpoints are useless when the overlap is low.

To tackle these challenges, we present a solution called the MS-Transformer. Drawing inspiration from [21], we introduce a random masking technique to encourage the network to learn correlations among different components. This approach has minimal impact on the results of downstream tasks while significantly enhancing computational efficiency. Specifically designed for registration tasks, we incorporate a masking mechanism during the subsampling process in the backbone network to reduce the number of points. Additionally, we utilize sparse self-attention to compute internal correlations and positional relationships within the point cloud, thereby reducing the computational cost of the transformer. By leveraging structural optimization and high-quality superpoint matching, our method offers several advantages. It enhances the number of inlier correspondences, resulting in improved registration robustness and accuracy, all without relying on RANSAC. Consequently, our approach surpasses the speed of RANSAC and other comparable methods while maintaining comparable accuracy levels. Extensive experiments on indoor benchmarks have demonstrated the effectiveness of our method. Our main contributions are:

- We present a fast and accurate point cloud registration method that relies on point correspondences, resulting in a significant improvement in the inlier ratio.
- We introduce a registration transformer based on sparse attention, which effectively learns the correlation between superpoints in registration scenarios.
- Through evaluation of various datasets, we demonstrate good registration accuracy and efficiency. Our method achieves well-aligned results even with small amounts of data.

II. Related Works

A. Correspondence-based Registration

The correspondence-based matching method first extracts features from the raw coordinate data of the source point cloud and the target point cloud, or directly uses the raw coordinate data as features. Secondly, for each coordinate point in the source point cloud, the feature similarity is calculated in a point-to-point manner using the features extracted from both point clouds. Then, the feature similarities are sorted and the point pair with the highest similarity is selected as the matching point pair for each point in the source point cloud to the target point cloud. Finally, the three-dimensional rigid body transformation matrix is solved using the matching point pairs based on singular value decomposition. In order to achieve higher registration accuracy, the above solution process is usually carried out iteratively, and the three-dimensional rigid body transformation estimated from the previous iteration will be used as the initial transformation for the next iteration. The iterative closest point [22] algorithm was the most basic and classic method for 3D point cloud registration. Later works improved the robustness of the algorithm to some extent by manually designing features to describe local geometric shapes [23], [24]. The above methods are all traditional methods and have relatively average registration accuracy in registration scenes where there is overlap in content. After the widespread application of deep learning, convolutional networks are used as mapping functions to adaptively extract high-dimensional features from point cloud coordinate information, while using real labels as strong constraints in the network training process. The earliest deep learning algorithm DCP [17] replaced the hand-designed feature extractor in the traditional method with DGCNN [20] and mimics the algorithm flow of ICP [25] for deep nearest neighbor point iterative calculation, which improves the registration performance compared to the traditional method. PRNet [19] improves on this basis. DeepGMR [26] used a Gaussian mixture model for deep learning registration and implements a ”point-distribution” matching strategy. D3Feat [9] performs well in indoor scenes, utilizing the point cloud feature extraction backbone network KPConv [27] and redesigning the detector and descriptor. PREDATOR [11] combines attention mechanisms to establish connections within and between point clouds and mainly considers point cloud registration in low-overlap situations based on D3Feat. Geotransformer [28] embeds geometric information into attention, resulting in further improvement.

B. Global Features-based Registration

Registration methods based on global features estimate the 3D rigid transformation parameters by utilizing the global information of the entire point cloud region, including both overlapping and non-overlapping areas. These methods are relatively new and all based on deep learning algorithms. Their advantage lies in the ability to avoid point correspondences mismatches caused by local receptive fields, insignificant geometric features, or strong noise interference by utilizing the global information of the point cloud. PointNetLK [29] was the first to propose a deep learning-based global feature registration method, utilizing PointNet [30] as the backbone network. PCRNet [31] improved upon PointNetLK in terms of noise sensitivity. FMR [32] restored the coordinate information of the input point cloud by the decoder and performs global feature registration. REGTR [33] is an end-to-end point cloud registration model that utilizes transformers to predict the

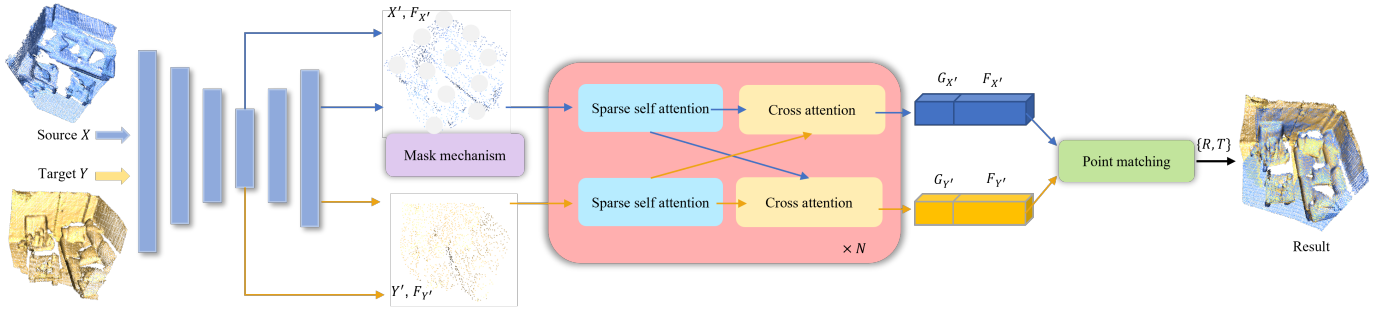


Fig. 2. We use KPConv as the backbone to perform point downsampling and feature extraction. The points obtained from the lowest layer of downsampling will undergo mask operation to reduce the number of superpoints. Then, the obtained superpoints and their features are fed into the alternating sparse self-attention layer and cross-attention layer for processing, and finally, point matching is performed to obtain the pose transformation matrix.

probability of each point in the overlapping area and its corresponding position in other point clouds, directly obtaining the required rigid transformation. Although methods based on global features solve the problem of local feature descriptors' inability to obtain global information or the high cost of obtaining global information through attention mechanisms, they do not consider the influence of non-overlapping areas on feature extraction in point clouds, and the global feature extraction process can be affected.

C. Registration Datasets

In the field of 3D point cloud registration, there are various public datasets available for different scenarios. For instance, for synthetic scenes, commonly used public datasets include the ModelNet40 [34] dataset and the ShapeNet [34] dataset. For indoor scenes, commonly used public datasets include the 7Scenes dataset and the 3DMatch [14] dataset. For outdoor scenes, commonly used public datasets include the KITTI [35] dataset, and the Oxford dataset [36]. However, at present, the availability of datasets for low-overlap registration is limited, with only the 3DMatch dataset being available. Furthermore, most existing methods are primarily designed for high-overlap point cloud registration. This indicates that there is still considerable potential for improvement in the field of low-overlap point cloud registration. It would be beneficial to propose additional datasets tailored to different scenarios to address this gap.

III. Method

Problem Definition: Given two point clouds:

$$X = \left\{X^i \in \mathbb{R}^3 | i = 1, \cdots, N\right\}, \tag{1}$$

$$Y = \left\{Y^i \in \mathbb{R}^3 | i = 1, \cdots, M\right\}, \tag{2}$$

which we denote as the source and target. The objective of point cloud registration is to recover the unknown rigid transformation consisting of a rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$ that aligns X to Y. The specific transformation can be solved by the following formula:

$$R^*, T^* = argmin R, T \sum_{l=1}^{N} ||Rx_l + t - \phi(x_l, Y)||^2, \tag{3}$$

we need to first establish point correspondence between two point clouds, and then estimate the alignment transformation.

A. MS-Transformer

Point sampling and masking. We adopt the KPConv-FCNN [27] model as the backbone network for point cloud downsampling, which accompanies point features during the downsampling process. The reason for downsampling point clouds is that a large number of points in point cloud registration are useless and can be matched using coarser-level correspondences. Over-clustered points may not produce good results. Similar to [18], [28], We refer to the sampled points at the lowest level as superpoints. Unlike previous works, we adopt the idea from [21] and perform further masking by randomly masking 50% of the superpoints. The remaining superpoints in the source and target point clouds are referred to as S' and T', respectively, and their corresponding learning features are denoted as $F^{s'} \in \mathbb{R}^{s' * d}$ and $F^{T'} \in \mathbb{R}^{T' * d}$. Let x_i and f_i represent the i-th point and feature in the point cloud, and the convolutional kernel g in x is defined as:

$$(F_{in} * g) \sum_{x_i \in N_x} g\left(x_i - x\right) f_i, \tag{4}$$

where N_x is the radius domain of point x, and x_i is a support point in this domain. In addition to the coarsest separation rate of superpoints, referring to the structures of [9], [37], we also added dense points corresponding to the first layer of downsampling, which can be used to construct a local patch around each superpoint using a point-to-point node grouping strategy.

Sparse self-attention. We designed a sparse self-attention structure that incorporates geometric spatial structure to learn the feature space and global correlations between superpoints in a point cloud. The attention mechanism involves huge computational costs when dealing with a large number of inputs [38]. Reducing attention computation has been a problem that researchers have been studying. Although point clouds contain a large number of points, most of them are useless in the case of low overlap. Their involvement in the calculation will instead affect the subsequent results. The extracted superpoints only need to establish local relationships with nearby points and global relationships with some distant points.

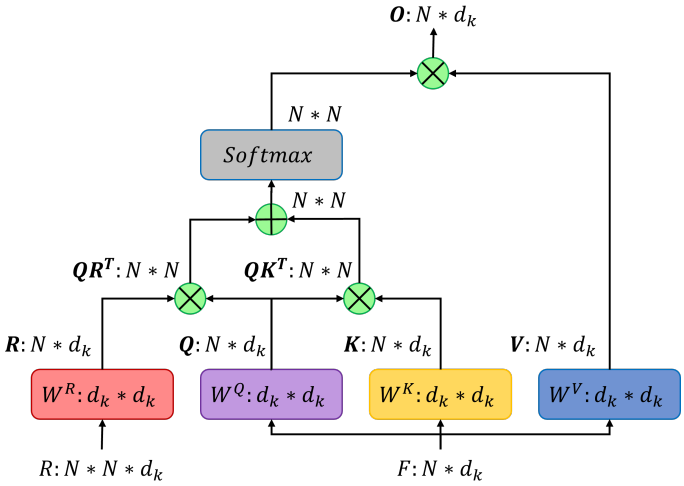


Fig. 3. Self-attention with spatial embedding.

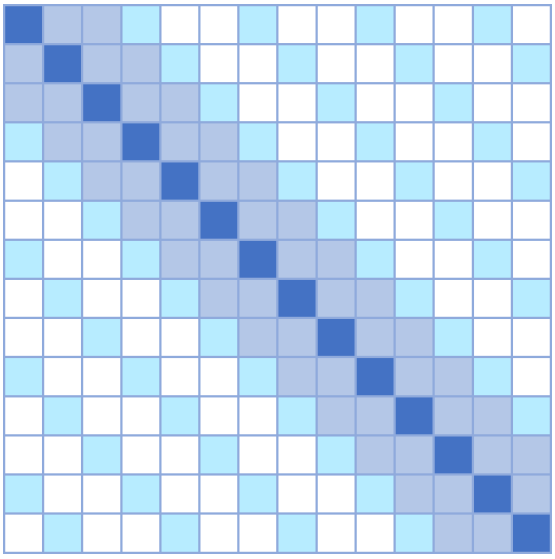


Fig. 4. sparse-attention only focuses on k surrounding points and step size.

The specific design is shown in the figure. We first encode a triplet with angles to replace the original position encoding for a learnable replacement [28]. We use Euclidean distance to measure the distance between two superpoints, $p_{i,j}$. Then the position encoding for a pair of points can be calculated as:

$$\begin{cases} r^D_{i,j,2k} = \sin \frac{d_{i,j}/\sigma_d}{10000^{2k/d_k}} \\ r^D_{i,j,2k+1} = \cos \frac{d_{i,j}/\sigma_d}{10000^{2k/d_k}}, \end{cases} \tag{5}$$

where d_k is the feature size and σ_d is the hyperparameter controlling the distance change. After obtaining the distance between two points, the algorithm for triplets is the same. Next, we introduce sparse attention [39] by applying sparse connections on K and V through the sparse features K_{S_i} and V_{S_i}, which reduces the complexity of QK^T. Then, we obtain the feature results for that moment and merge all the results together to obtain the final $Atten(X, S)$ [40]. This way, the attention has the characteristics of local dense correlation and remote sparse correlation.

$$Atten\left(X, S\right) = \left(a\left(x_i, S_i\right)\right)_{i \in \{1,\ldots,n\}}, \tag{6}$$

$$a\left(x_i, S_i\right) = softmax\left(\frac{\left(W_q x_i\right) K^T_{S_i}}{\sqrt{d}}\right) V_{S_i}, \tag{7}$$

$$K_{S_i} = \left(W_k x_j\right)_{i_i} V_{S_i} = \left(W_v x_j\right)_{i_i}. \tag{8}$$

Feature cross-attention. The cross-attention module is widely used in point cloud registration tasks [28], [33], [41], for feature exchange between the source and target point clouds. After obtaining the feature matrices F^X and F^Y through self attention, calculate the cross attention feature matrix Z^x of X' with Y' feature:

$$Z^X_i = \sum_{j=1}^{|Y'|} A_{i,j}(F^Y_j W^V), \tag{9}$$

$$a_{i,j} = \frac{(F^X_i W^Y)(F^Y_i W^Q)^T}{\sqrt{d_t}}. \tag{10}$$

$a_{i,j}$ are the feature correlation scores between F^x and F^y, and $A_{i,j}$ is obtained by row-wise softmax on $a_{i,j}$. The cross-attention feature of Y can be obtained in the same way. The self-attention module encodes a geometric structure that remains unchanged within each point cloud, while the cross-attention module interacts between two point clouds to obtain geometric consistency for modeling [42]. The resulting mixed features are invariant to the transformation.

B. Point Matching

After alternately passing through n rounds of S-transformer, according to [28], we first normalize the superpoint features $F^{X'}$ and $F^{Y'}$ to a unit hypersphere, and calculate the similarity of point pairs using the Gaussian correlation matrix. Then, we further perform double normalization to suppress ambiguous matches and effectively reduce incorrect matches. Finally, we select the top k pairs of superpoints with the highest correlation as the coarse correspondence set for subsequent registration $C' = \left\{(x'_i, y'_j) | x'_i \in X', y'_i \in Y'\right\}$.

After obtaining C', the points $\hat{X}$ and $\hat{X}$ with higher density after downsampling through the first layer can be assigned to the corresponding superpoints, and calculated using the Sinkhorn algorithm [43] in optimal transmission theory. The specific method is as follows: Firstly, a hyperpoint is given, and its corresponding point group is $\hat{G}^x_i \subseteq \hat{X}$, and the feature groups associated with it can be defined as $\hat{G}^{F_x}_i \subseteq \widehat{F_X}$. The similarity of feature groups between X and Y can be calculated as $\widehat{S}_l = \hat{G}^{F_x}_i (\hat{G}^{F_y}_j)^T / \sqrt{\hat{c}}$, with $\hat{c}$ the feature dimension. Then add a new row and column to the similarity matrix calculation, which is used to fill in the learnable parameter α, and iteratively calculate the Sinkhorn algorithm to obtain the normalized similarity matrix. Next, remove the last row and column, so that entries with top-k confidence on both rows and columns are formed to form the point correspondence set

C_l. Ultimately forming a globally dense point correspondences $\mathbb{C} = \bigcup_{l=1}^{|C|} \mathbb{C}_l$.

C. Loss Function

We use two loss functions to monitor the ground truth relationships:$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_p$ which $\mathcal{L}_s$ is superpoint matching loss and $\mathcal{L}_p$ is point matching loss.

Superpoint Matching Loss. We use circle loss [44] for superpoint matching, measuring different ground conditions and superpoint correspondence by using the overlap rate near the superpoints [9], [11]. Select a set of superpoints from Y, and if there is more than 10% overlap with the selected superpoints in X, it is considered a positive sample. Otherwise, it is called a negative sample [45]. The set with at least one positive sample is called set N. For each X in the set, the positive samples in Y are called $\mathcal{E}_p^i$ and negatives as $\mathcal{E}_n^i$. So the superpoint matching loss can be defined as:

$$\mathcal{L}_s^X = \frac{1}{N} \sum_{g_i^X} log[1 + \sum_{g_i^Y \in \mathcal{E}_p^i} e^{\lambda_i^j \beta_X^{i,j} (d_i^j - \Delta p)} \sum_{g_k^Y \in \mathcal{E}_n^i} e^{\beta_n^{i,k} (\Delta n - d_i^k)}], \tag{11}$$

where d is the distance in the feature space, λ Indicates the overlap ratio between two patches, β Is the positive and negative weight of each sample [46]. Set margin hyper-parameters Default setting is $\Delta p = 0.1$ and $\Delta n = 0.1$. In this way, patches with high overlap become more important [47]. The reverse loss $\mathcal{L}_s^Y$ on Y is computed in the same way. For total superpoint matching loss is $\mathcal{L}s = (\mathcal{L}_s^X + \mathcal{L}_s^Y)/2$.

Point Matching Loss. We use negative logarithmic likelihood loss on the allocation matrix of superpoints [28], [41]. During the training process, we do not use the predicted correspondence, but rather randomly sample the C_g ground truth superpoint correspondence $\{\hat{C}_i^*\}$, M_i is the set of ground-truth point correspondences. Using P_i and Q_i respectively to represent the set of mismatched points in two patches, we can compute the individual point matching loss as:

$$\mathcal{L}p, i = -\sum_{(u,v) \in M_i} \log \bar{z}_{u,v}^i - \sum_{u \in P_i} \log \bar{z}_{u,m_i+1}^i - \sum_{v \in Q_i} \log \bar{z}_{n_i+1,v}^i. \tag{12}$$

The total point matching loss is the average of all individual losses $\mathcal{L}p = \frac{1}{C_g} \sum_{i=1}^{C_g} \mathcal{L}_{p,i}$

IV. Experiment

A. Dataset

We evaluated the performance of our method on indoor 3DMatch [14] and 3DLoMatch datasets. The 3DMatch dataset is divided into three sets, namely training, validation, and test sets, each consisting of 46, 8, and 8 scenes, respectively. We used preprocessed data from Predator and followed its split, where the 3DLoMatch dataset includes only scenes with overlap ratios between 10-30%. Each input point cloud in the dataset contains no more than 30,000 points and was down-sampled using the KPConv [27] backbone and augmented with small rigid perturbations during training.

B. Evaluation Metrics

Similar to most of the current point cloud registration works, we evaluated the performance of our methods using the standard metrics of 3DMatch, which mainly includes three metrics [9], [11]: 1)registration recall (RR), the most important metric that calculates the root mean square error of the transformed correspondences to measure the success of registration (defined as the RMSE of corresponding point pairs should be less than 0.2); 2)feature matching recall (FMR), which measures the score of point cloud pairs to judge whether a pair of point clouds is suitable for registration and evaluate the potential success of registration; 3)inlier ratio (IR), which is the ratio of the number of point pairs in the overlapping area used for matching to all the point pairs, and if the distance between two points is less than 10 cm in the true transformation, it is considered as an inlier.

C. Implementation

We conducted experiments on our method in the PyTorch environment, trained for 20 epochs, using a single NVIDIA RTX3090 GPU with 24GB memory, and an Intel 12900k CPU. The training took approximately 24 hours. For data augmentation, we added 0.005 noise perturbation and proportional rotation perturbation, with a batch size of 1. We used an initial learning rate of 10^{-4}, weight decay of 10^{-6}, and the learning rate was decayed by a factor of 0.05 after each epoch. The backbone network underwent two down-samplings, and the transformer alternated between four layers of self-attention and cross-attention. We used a multi-head attention mechanism with 4 heads and 32 groups of superpoints. For the superpoint matching loss, we calculated it using up to $n_p = 64$ positive pairs and selected the top_k=3, using the Sinkhorn algorithm for 100 iterations. When compared to RANSAC, we performed 1000 iterations.

D. Registration Results.

Comparison with recent methods. We compared our method with some recent methods and feature-based registration methods. In Table 1, we show the matching and registration results of 3DMatch and 3DLoMatch using 5000 points/correspondence. It can be seen that the FMR [32] ratio of recent methods is high, and the registration recall rate is also high on the 3DMatch dataset. Our method is slightly lower than CoFiNet [18], but in the low overlap 3DLoMatch, the recall rate of other methods decreases significantly, and our method is in a leading position. In terms of the inlier ratio, our method outperforms other methods by a large margin, reaching more than 20 points, which is in line with our expectations and indicates that our method is better at dealing with difficult situations.

Scenes results. We also conducted separate experiments on different scenes within the 3DMatch and 3DLoMatch datasets, which included a total of 8 scenes. In the 3DMatch dataset, FMR was generally high, and the Hotel_1 scene had the highest average inlier ratio, while the Home_1 scene had the highest registration recall rate, with a difference of up to 12%

TABLE I
RESULTS ON THE 3DMATCH AND 3DLOMATCH DATASETS

#samples	3DMatch 5000	3DLoMatch 5000
Feature Match Recall(%)		
3DSN [48]	95.0	63.6
FCGF [49]	97.4	76.6
D3Feat [9]	95.6	67.3
Predeator [11]	96.6	71.7
SpinNet [50]	97.6	75.3
YOHO [51]	**98.2**	79.4
CoFiNet [52]	98.1	83.1
Sparse-transformer(ours)	98.1	**86.3**
Inlier Ratio(%)		
3DSN [48]	36.0	11.4
FCGF [49]	56.8	21.4
D3Feat [9]	39.0	13.2
Predeator [11]	49.9	20.0
SpinNet [50]	47.5	20.5
YOHO [51]	64.4	25.9
CoFiNet [52]	49.8	24.4
Sparse-transformer(ours)	**72.8**	**53.3**
Registration Recall(%)		
3DSN [48]	78.4	33.0
FCGF [49]	85.1	40.1
D3Feat [9]	81.6	37.2
Predeator [11]	88.3	54.2
SpinNet [50]	88.6	59.8
YOHO [51]	90.8	65.2
CoFiNet [52]	**89.3**	67.5
Sparse-transformer(ours)	89.1	**71.3**

TABLE II
RESULTS IN DIFFERENT SCENES ON THE 3DMATCH AND 3DLOMATCH DATASETS

	3DMatch			3DLoMatch		
SCENES	FMR	IR	RR	FMR	IR	RR
Kitchen	0.994	0.812	0.921	0.928	0.563	0.826
Home_1	0.994	0.817	**0.98**	0.903	0.502	0.678
Home_2	0.971	0.753	0.824	0.874	0.558	0.721
Hotel_1	**1.000**	**0.834**	0.973	**0.963**	**0.655**	**0.890**
Hotel_2	0.990	0.802	0.91	0.861	0.532	0.674
Hotel_3	**1.000**	0.831	0.855	0.796	0.578	0.690
Study	**1.000**	0.761	0.902	0.771	0.420	0.523
MIT_lab	0.949	0.707	0.911	0.806	0.454	0.700

compared to the worst-performing scene. In the 3DLoMatch dataset, all indicators performed best in the Hotel_1 scene. However, from the original data, the kitchen and study scenes had the largest number of point pairs, and we found that the quality of the matching results did not depend solely on the number of point pairs, but rather on the quality of the matched points, as poor feature points could actually have a negative impact on the results.

TABLE III
THE INFLUENCE OF ROTATION ON SELF-ATTENTION

Model	3DMatch original	3DMatch rotated	3DLoMatch original	3DLoMatch rotated
(a)self-attention w/ACE	87	85.1	65.7	64.3
(b)self-attention w/RCE	86.8	86.8	65.1	65.1
(c)sparse self-attention	**89.1**	**88.9**	**71.3**	**71**

TABLE IV
RUNTIME COMPARISON.

Methods	Data(s)	Model(s)	Total(s)
Lepard [53]	0.444	0.051	0.495
ConFiNet [52]	0.211	0.115	0.326
Geotransformer	0.194	0.075	0.269
Ours	**0.153**	**0.032**	**0.185**

Rotation invariance. We evaluated the rotational invariance of different positional embeddings in Table 3. During training, we introduced rotation perturbations [11], [28]. In different experiments, the performance of the self-attention with absolute coordinate embeddings (a) was significantly reduced, indicating that it failed to deal with variance changes due to transformations. However, although the performance of self-attention with relative coordinate embeddings (b) was not as good as that with absolute position embeddings in normal situations [54], the model's results remained unchanged after adding rotation perturbations, indicating that relative position embeddings had no effect on the model [55]. Our method (c) demonstrated good rotational invariance relative to perturbations and had an effective effect on encoding spatial structure.

Experimental runtime. We ran the methods listed in the table on a machine equipped with a single Nvidia RTX 3090 GPU and an Intel 12900K CPU. All models were tested with a batch size of 1 and without CPU parallelism. The final time reported is an average taken over 1623 point cloud pairs in the 3DMatch test set. The "Data" column reports the time it takes to prepare the data, while the "Model" column reports the time it takes to generate descriptors from the prepared data. Although all of these methods use KPConv as the backbone, our model benefits from the masking mechanism and sparse attention, which significantly reduces computation time and alleviates the computational pressure of the transformer. Our model can also be experimented with on larger point cloud datasets in the future.

V. CONCLUSION

We present the MS-Transformer, a deep model specifically designed for low-overlap point cloud registration tasks. The model's core features a sparse attention mechanism based on a masked transformer, complemented by spatial positional encoding and sparse self-attention. These components establish internal relationships and geometric consistency within the source point cloud. Moreover, cross-attention is employed to establish connections between the source and target point clouds, reducing the computational load of the transformer while enhancing registration accuracy without compromising speed. However, the issue of losing key matching points still persists due to the reduction in the number of matching points. In the future, our goal is to devise improved methods for selecting matching points and integrate point cloud registration with the utilization of labels from other visual tasks.

REFERENCES

[1] D. Cao et al., "Future Directions of Intelligent Vehicles: Potentials, Possibilities, and Perspectives," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 1, pp. 7-10, March 2022, doi: 10.1109/TIV.2022.3157049.

[2] H. Wang et al., "A Pathway Forward: The Evolution of Intelligent Vehicles Research on IEEE T-IV," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 4, pp. 918-928, Dec. 2022, doi: 10.1109/TIV.2022.3215784.

[3] T. -H. Chen and T. S. Chang, "RangeSeg: Range-Aware Real Time Segmentation of 3D LiDAR Point Clouds," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 1, pp. 93-101, March 2022, doi: 10.1109/TIV.2021.3085827.

[4] A. Venon, Y. Dupuis, P. Vasseur and P. Merriaux, "Millimeter Wave FMCW RADARs for Perception, Recognition and Localization in Automotive Applications: A Survey," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 3, pp. 533-555, Sept. 2022, doi: 10.1109/TIV.2022.3167733.

[5] B. Weng, L. Capito, U. Ozguner and K. Redmill, "Towards Guaranteed Safety Assurance of Automated Driving Systems With Scenario Sampling: An Invariant Set Perspective," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 3, pp. 638-651, Sept. 2022, doi: 10.1109/TIV.2021.3117049.

[6] K. Samal, H. Kumawat, P. Saha, M. Wolf and S. Mukhopadhyay, "Task-Driven RGB-Lidar Fusion for Object Tracking in Resource-Efficient Autonomous System," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 1, pp. 102-112, March 2022, doi: 10.1109/TIV.2021.3087664.

[7] L. Luo, S. -Y. Cao, Z. Sheng and H. -L. Shen, "LiDAR-Based Global Localization Using Histogram of Orientations of Principal Normals," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 3, pp. 771-782, Sept. 2022, doi: 10.1109/TIV.2022.3169153.

[8] T. Gao, H. Pan and H. Gao, "Monocular 3D Object Detection With Sequential Feature Association and Depth Hint Augmentation," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 2, pp. 240-250, June 2022, doi: 10.1109/TIV.2022.3143954.

[9] 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 CVPR, pages 6359–6367, 2020.

[10] 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 CVPR, pages 15859–15869, 2021.

[11] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In CVPR, pages 4267–4276, 2021.

[12] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In CVPR, pages 5545–5554, 2019.

[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] Andy Zeng, Shuran Song, Matthias Nie?ner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In CVPR, pages 1802–1811, 2017.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proc. CVPR, pages 3354–3361, 2012.

[16] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[17] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In Proc. ICCV, pages 3523–3532, 2019.

[18] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust point cloud registration. arXiv preprint arXiv:2110.14076, 2021.

[19] Yue Wang and Justin Solomon. Prnet: self-supervised learning for partial-to-partial registration. In NeurIPS, pages 8814–8826, 2019.

[20] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. ACM TOG, 38(5), 2019.

[21] He K, Chen X, Xie S, et al. Masked autoencoders are scalable vision learners[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 16000-16009.

[22] Paul J. Besl and Neil D. McKay. A method for registration of 3d shapes. volume 14, pages 239–256, 1992.

[23] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3d registration. In IEEE International Conference on Robotics and Automation (ICRA), pages 3212–3217, 2009.

[24] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In International Conference on Intelligent Robots and Systems, pages 3384–3391, 2008.

[25] Szymon Rusinkiewicz. A symmetric objective function for icp. ACM Trans. Graphics, 38(4):1–7, 2019.

[26] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In Proc. ECCV, pages 733–750, 2020.

[27] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Franc?ois Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In ICCV, pages 6411–6420, 2019.

[28] Qin Z, Yu H, Wang C, et al. Geometric transformer for fast and robust point cloud registration[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 11143-11152.

[29] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & efficient point cloud registration using pointnet. In Proc. CVPR, pages 7163–7172, 2019.

[30] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proc. CVPR, pages 652–660, 2017.

[31] Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Rangaprasad Arun Srivatsan, Simon Lucey, and Howie Choset. Pcrnet: Point cloud registration network using pointnet encoding. arXiv preprint arXiv:1908.07906, 2019.

[32] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Featuremetric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In Proc. CVPR, pages 11366–11374, 2020.

[33] Yew Z J, Lee G H. Regtr: End-to-end point cloud correspondences with transformers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 6677-6686.

[34] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In CVPR, 2015.

[35] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In ECCV,pages 607–623, 2018.

[36] Maddern W, Pascoe G, Linegar C, et al. 1 year, 1000 km: The Oxford RobotCar dataset[J]. The International Journal of Robotics Research, 2017, 36(1): 3-15.

[37] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer International Publishing, 2015: 234-241.

[38] X. Tang et al., "Prediction-Uncertainty-Aware Decision-Making for Autonomous Vehicles," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 4, pp. 849-862, Dec. 2022, doi: 10.1109/TIV.2022.3188662.

[39] Child R, Gray S, Radford A, et al. Generating long sequences with sparse transformers[J]. arXiv preprint arXiv:1904.10509, 2019.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ?ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, pages 5998–6008, 2017

[41] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In CVPR, pages 4938–4947, 2020.

[42] S. Ansari, F. Naghdy and H. Du, "Human-Machine Shared Driving: Challenges and Future Directions," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 3, pp. 499-519, Sept. 2022, doi: 10.1109/TIV.2022.3154426.

[43] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics, 21(2):343–348, 1967.
[44] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In CVPR, pages 6398–6407, 2020.
[45] E. de Gelder et al., "Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles: An Object-Oriented Framework," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 2, pp. 300-314, June 2022, doi: 10.1109/TIV.2022.3144803.
[46] X. Zhu et al., "Interaction-Aware Cut-In Trajectory Prediction and Risk Assessment in Mixed Traffic," in IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 10, pp. 1752-1762, October 2022, doi: 10.1109/JAS.2022.105866.
[47] L. Yang et al., "Collective Entity Alignment for Knowledge Fusion of Power Grid Dispatching Knowledge Graphs," in IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 11, pp. 1990-2004, November 2022, doi: 10.1109/JAS.2022.105947.
[48] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In CVPR, 2019.
[49] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In ICCV, 2019.
[50] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In CVPR, pages 11753–11762, 2021.
[51] Wang H, Liu Y, Dong Z, et al. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors[C]//Proceedings of the 30th ACM International Conference on Multimedia. 2022: 1630-1641.
[52] Yu H, Li F, Saleh M, et al. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration[J]. Advances in Neural Information Processing Systems, 2021, 34: 23872-23884.
[53] Li Y, Harada T. Lepard: Learning partial point cloud matching in rigid and deformable scenes[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 5554-5564.
[54] H. Mo, Y. Meng, F. -Y. Wang and D. Wu, "Interval Type-2 Fuzzy Hierarchical Adaptive Cruise Following-Control for Intelligent Vehicles," in IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 9, pp. 1658-1672, September 2022, doi: 10.1109/JAS.2022.105806.
[55] X. Li, H. Duan, Y. Tian and F. -Y. Wang, "Exploring Image Generation for UAV Change Detection," in IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 6, pp. 1061-1072, June 2022, doi: 10.1109/JAS.2022.105629.