GERA: Geometric Embedding for Efficient Point Registration Analysis

Geng Li[1,2], Haozhi Cao[1], Mingyang Liu[2], Shenghai Yuan[1], Jianfei Yang[1,*]

***Abstract*—Point cloud registration aims to provide estimated transformations to align point clouds, which plays a crucial role in pose estimation of various navigation systems, such as surgical guidance systems and autonomous vehicles. Despite the impressive performance of recent models on benchmark datasets, many rely on complex modules like KPConv and Transformers, which impose significant computational and memory demands. These requirements hinder their practical application, particularly in resource-constrained environments such as mobile robotics. In this paper, we propose a novel point cloud registration network that leverages a pure MLP architecture, constructing geometric information offline. This approach eliminates the computational and memory burdens associated with traditional complex feature extractors and significantly reduces inference time and resource consumption. Our method is the first to replace 3D coordinate inputs with offline-constructed geometric encoding, improving generalization and stability, as demonstrated by Maximum Mean Discrepancy (MMD) comparisons. This efficient and accurate geometric representation marks a significant advancement in point cloud analysis, particularly for applications requiring fast and reliability.**

***Index Terms*—Point cloud registration, Point cloud representative learning**

I. Introduction

Point cloud registration is a fundamental problem in 3D scene understanding and robotics. The goal is to acquire the point-wise transformation matrix that aligns two point clouds, which plays a vital role in various downstream tasks (e.g., motion estimation [1], [2], 3D scene reconstruction [3], SLAM [4], [5], and surgical navigation [6]).

Recent non-learning-based methods for non-rigid registration perform well but are hindered by high inference time, limiting real-time use. Learning-based methods, which eliminate complex optimization at test time, improve efficiency but still fall short of real-time performance, necessitating further enhancements for practical deployment [7], [8]. A typical example is the point cloud registration of organs in surgical robots [9], [10] as shown in Fig. 1. On the one hand, the registration algorithm, in this case, needs to be robust and highly accurate to guarantee the success of the surgery. On the other hand, it must satisfy the real-time inference requirement even with rather limited computational resources (e.g., embedded devices). Therefore, developing accurate and efficient learning-based point cloud registration methods is significant, bridging the gap between benchmarking and real-life deployment.

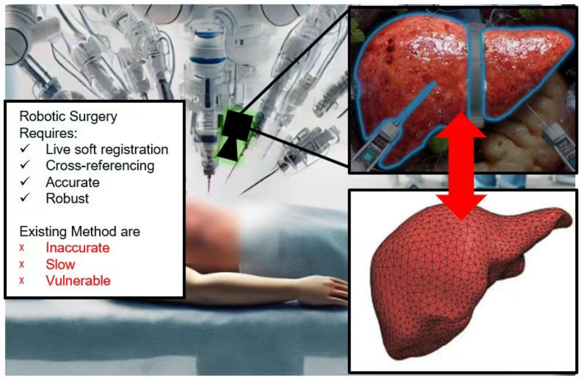

Fig. 1: In robot-assisted surgery, real-time analysis and registration of target organs are required, which presents new challenges for robustness and efficiency.

Existing learning-based solutions [7], [8] rely on complex local geometric feature extractors that extract usable registration features from raw 3D points [11], [12]. Despite their effectiveness, these powerful extractors (e.g., KPConv [11] and Transformer [12]) usually contain masses of trainable parameters, bringing significant computational expenses during both the training and inference stages. As the model becomes increasingly complex, its performance improvements fall short of meeting the demands for real-time processing.

We revisited recent learning-based registration methods and found that their primary efficiency bottleneck stems from feature extraction and online local construction modules [7], [8], [13]. Drawing inspiration from non-learning methods [14], we propose replacing traditional positional information with geometric features, to optimize registration performance. Our approach constructs information-rich geometric-encoded inputs, which offer two key advantages. First, by using efficient encoding algorithms, geometric 3D representations can be extracted without learnable modules, reducing the computational load during both offline training and online inference. Second, these geometric representations are more informative than raw 3D points, leading to better performance even with simpler learnable modules.

Based on the above analysis, in this paper, we propose GERA, a method for GEometric embedding for leaRning-

[1]Nanyang Technological University, Singapore.
[2]Shandong University, China.
[*]Corresponding author: `jianfei.yang@ntu.edu.sg`

based efficient point registrAtion with a lightweight trainable network, improving both the effectiveness and efficiency of registration. Given raw point clouds as input, we first construct a 3D descriptor for each point by forming a fully connected graph using this point and its neighboring points. The edges of the graph represent the distances between every pair of points. We conducted a kernel-based statistical analysis using Maximum Mean Discrepancy(MMD) [15] to demonstrate the stability and robustness of our geometric embedding encoding. which is a kernel-based statistical measure employed to analyze the similarity of encoded embeddings between different input samples. By leveraging the MMD, we demonstrated that the stability of geometric information encoding features is superior to previous state-of-the-art learning-based solutions. Additionally, our descriptors can be constructed offline, significantly reducing training and inference time. Following prior work that employs MLP architectures for point cloud analysis and registration, we utilize this simple network architecture to effectively leverage the informative descriptors, thereby significantly surpassing previous complex learning-based solutions. Our experiment results show that the inference speed was increased by 22x, resulting in a 115% improvement in prediction accuracy compared to the existing state-of-the-art solution.

Our contributions can be summarized as follows:

- We propose an offline method that efficiently and accurately constructs geometric information from point clouds.
- We are the first to replace 3D coordinate input with geometric encoding information in point cloud processing.
- We surpassed the SOTA by 12.5% while using only 3% of the computation time required by existing methods.

II. Related work

A. Feature Encoding

Feature encoding for point cloud data began with PointNet [16], which utilized rotation-invariance and an MLP architecture but lacked the ability to capture local geometric information. This was improved by PointNet++ [13], which introduced hierarchical feature extraction from local to global levels, and DGCNN [17], which considered the relative distances between neighboring points. KPConv [11] further advanced this by extracting features within a spherical range, similar to convolution. In point cloud registration, methods like GeoTransformer [8] and Lepard [7] adopted transformer architectures to effectively model local geometric information, though with increased computational overhead.

B. Non-learning-based Methods

The Iterative Closest Point (ICP) [18] algorithm based on the least squares method relies heavily on initialization and converges slowly. Coherent Point Drift [19] (CPD) formulates the registration task as a probability density estimation problem, iteratively updating point correspondences and non-rigid transformation parameters using the expectation-maximization technique. However, CPD requires multiple iterations and is sensitive to occlusions and outliers. Methods employing Graph-Laplacian regularization [20] and context-aware Gaussian fields (SCGF) [21] improve correspondence and transformation estimation but depend on high-quality assignment initialization. Bayesian Coherent Point Drift (BCPD) [22] enhances CPD's convergence through variational Bayesian inference but remains prone to local minima. The MR-RPM [23] method based on manifold regularization captures the intrinsic geometry of point sets using manifold regularization priors. PointSetReg [24] utilizes unsupervised clustering analysis, avoiding explicit point correspondences to improve robustness and efficiency. However, the heavy reliance of PointSetReg on cluster-level information may reduce precision when handling complex point clouds.

C. Learning-based Methods

FPT [6] utilizes a decoder composed of PointNet and fully connected layers to perform non-rigid registration on prostate point clouds, offering high efficiency but limited robustness due to the lack of a geometric construction module. Lepard [7] employs a transformer architecture to form attention heads between point clouds, extracting the point correspondence matrix, which is then input into algorithms such as N-ICP [14] for non-rigid registration. While Lepard provides comprehensive encoding of point cloud information, its slower processing speed limits practical applications. Similar to non-rigid registration, the problem of scene flow estimation has garnered increasing attention in recent years. PointPWC-Net [25] refines the flow iteratively at multiple scales, enhancing its ability to capture fine-grained motion details, but its computational complexity remains relatively high. BI-PointFlowNet [26] incorporates bidirectional learning for point clouds, enhancing accuracy and robustness, but it encounters difficulties when handling scenes with large displacements. DiffFlowNet [27] utilizes a powerful and complex diffusion model, achieving millimeter-level error in scene flow estimation, but it also introduces the challenge of prolonged inference time.

III. METHODOLOGY

Probelm Definition: In non-rigid registration, we are given a source point set $\mathbf{P}_{\mathcal{S}} = \{x_1, \dots, x_i, \dots, x_M\}$ and a target point set $\mathbf{P}_{\mathcal{T}} = \{y_1, \dots, y_j, \dots, y_N\}$, where $x_i, y_j \in \mathbb{R}^3$ represent the 3D coordinates of the points, and M, N are their respective counts. The goal is to estimate displacement vectors $\mathbf{D}_{\text{pred}} = [\mathbf{d}_{\text{pred}_1}, \dots, \mathbf{d}_{\text{pred}_M}]$, which deform each point $x_i \in \mathbf{P}_{\mathcal{S}}$ to align with $\mathbf{P}_{\mathcal{T}}$.

The deformed point set $\mathbf{P}'_{\mathcal{S}} = \{x'_1, \dots, x'_i, \dots, x'_M\}$ is computed as:

$$x'_i = x_i + f(x_i, \mathbf{D}_{\text{pred}}) = x_i + \mathbf{d}_{\text{pred}_i} + \epsilon(x_i), \qquad (1)$$

where x'_i is the deformed point in $\mathbf{P}'_{\mathcal{S}}$, and $\epsilon(x_i)$ represents a small adjustment term to enhance registration smoothness.

We propose GERA, which enhances point cloud registration by leveraging offline 3D geometric representations. By embedding neighbor information in a point-wise manner, our method achieves more stable encoding than 3D coordinates, as shown by MMD [15] analysis. Therefore, GERA can significantly surpass existing solutions with more lightweight learnable modules, as shown in Fig. 3

A. *Efficient Offline Geometric Representations*

Previous studies on local geometric information used 3D spatial coordinates of point clouds with online extraction, resulting in a computational complexity of $O(n)$, where n is the number of training epochs, demanding significant computational and memory resources. This paper instead proposes an offline geometric information construction method with $O(1)$ complexity, significantly alleviating the overhead during the training process. Based on our quantitative analysis with MMD, we further demonstrate that our proposed geometric representations possess superior generalizability and stability compared to others.

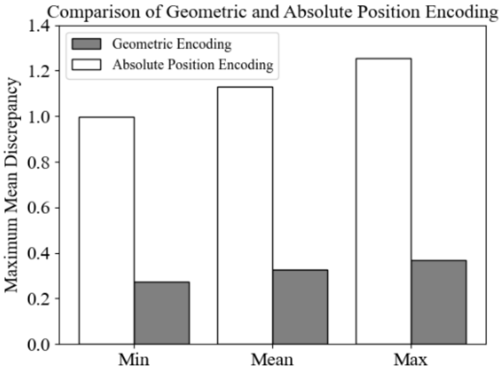

Fig. 2: Perform MMD analysis on the liver dataset using a batch size of 32 for both absolute position encoding and geometric encoding

1) Construction of Geometric Representations: Our method improves upon previous local geometric information construction by using a simple geometric prior. Traditional methods [8] compute distances and angles between points and neighbors, which are not directly comparable to MLP input because angles and distances represent different dimensions of information. By forming a triangle with the original point and its two neighbors, and defining it by side lengths, we reduce the geometric relationships to a consistent and usable format for the network, which can be formulated as:

$$\begin{aligned} \alpha &= \|\mathbf{P}_i - \mathbf{P}_j\|, & \beta &= \|\mathbf{P}_i - \mathbf{P}_k\|, \\ \gamma &= \|\mathbf{P}_j - \mathbf{P}_k\|, & \mathbf{G}_{(i,j,k)} &= \text{concat}(\alpha, \beta, \gamma), \end{aligned} \tag{2}$$

where $\mathbf{P}_i$, $\mathbf{P}_j$, $\mathbf{P}_k$ are the coordinates of points, and $\mathbf{G}_{(i,j,k)}$ donates the geometric information constructed for points i, j, k. concat$(\cdot)$ indicates the concatenation operation.

Using only two neighboring points for geometric construction forms the simplest fully connected graph, yielding limited information. To further enrich the embedded information, we consider each point's n neighbors as vertices of a fully connected graph, resulting in C_n^2 pairwise distances. This comprehensive modeling retains only the edge lengths, which, according to distance geometry and rigidity theory, uniquely determine a 3D structure [28]. This approach is widely used in structural chemistry and drug design [29], [30], which could be formulated as:

$$\mathbf{sc}^i \in \mathbb{R}^d = \text{concat}(\{\mathbf{G}_{(i,j,k)} \mid 1 \leq j < k \leq n\}), \tag{3}$$

where $\mathbf{sc}^i$ denotes the fully connected graph constructed from n neighboring points. d represents the total number of edges in the fully connected graph, C_n^2.

2) Analysis of Stability and Generalization: Non-rigid point cloud registration entails more complex transformations than rigid registration, resulting in flexible data distribution variations. Thus, a robust feature representation with strong generalization is essential. To assess the stability and generalization of geometric encoding, we performed a quantitative analysis based on MMD, a criterion widely used to measure discrepancy in domain adaptation [31]–[33].

Specifically, MMD measures the distance between samples in a high-dimensional feature space, with smaller distances indicating greater similarity and thus stronger generalization and stability. In this context, MMD can provide a quantitative assessment about the similarity of the encoded features between samples, expressed as:

$$\text{MMD}^2(\mathbf{P}_\mathcal{S}, \mathbf{P}_\mathcal{T}) = \|\mathbb{E}_{x \sim \mathbf{P}_\mathcal{S}}[\phi(x)] - \mathbb{E}_{y \sim \mathbf{P}_\mathcal{T}}[\phi(y)]\|_\mathcal{H}^2 , \tag{4}$$

where $\phi(x)$ is the mapping function that projects the x into a high-dimensional Reproducing Kernel Hilbert Space $\mathcal{H}$.

By expanding and simplifying equation (3), we can derive the computation formula for MMD:

$$\begin{aligned} \text{MMD}^2(\mathbf{P}_\mathcal{S}, \mathbf{P}_\mathcal{T}) = \mathbb{E}_{x,x' \sim \mathbf{P}_\mathcal{S}}[k(x, x')] + \mathbb{E}_{y,y' \sim \mathbf{P}_\mathcal{T}}[k(y, y')] \\ - 2\mathbb{E}_{x \sim P, y \sim Q}[k(x, y)], \end{aligned} \tag{5}$$

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \tag{6}$$

where x and x' represent two independent samples drawn from the distribution $\mathbf{P}_\mathcal{S}$, and $k(\cdot, \cdot)$ is the kernel function. Equation (5) specifies the Gaussian kernel function.

In this study, we employed identical encoder architectures for both 3D coordinate encoding and geometric encoding as inputs to MMD. As shown in Fig. 2, the MMD values produced by geometric encoding were significantly lower than those from position encoding, indicating that GERA's geometric encoding method is more stable and has stronger generalization capabilities. Moreover, the minimum, mean, and maximum MMD values for geometric encoding were 0.2736, 0.3178, and 0.3668, respectively, which are notably

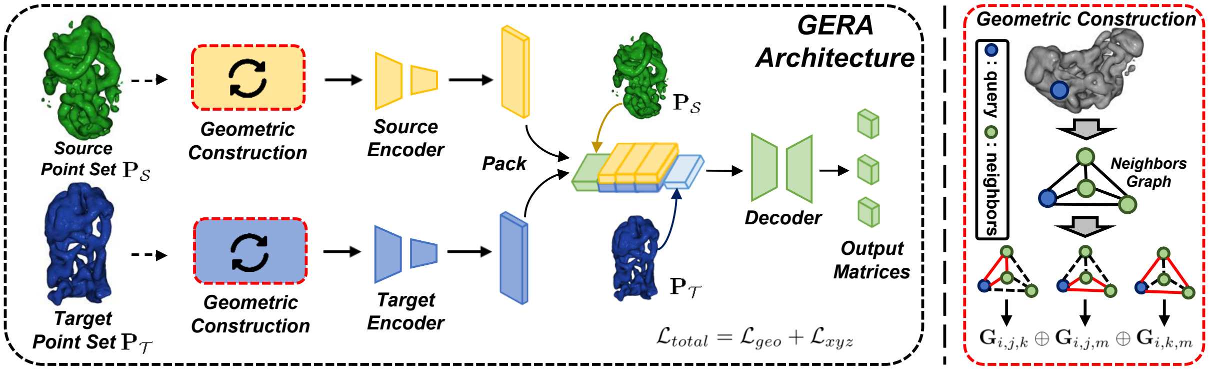
GERA
Architecture
Source
Point Set $\mathbf{P}_{\mathcal{S}}$
Geometric
Construction
Source
Encoder
Pack
$\mathbf{P}_{\mathcal{S}}$
Decoder
Output
Matrices
Target
Point Set $\mathbf{P}_{\mathcal{T}}$
Geometric
Construction
Target
Encoder
$\mathbf{P}_{\mathcal{T}}$
$\mathcal{L}_{total} = \mathcal{L}_{geo} + \mathcal{L}_{xyz}$
Geometric Construction
: query
: neighbors
Neighbors
Graph
$\mathbf{G}_{i,j,k} \oplus \mathbf{G}_{i,j,m} \oplus \mathbf{G}_{i,k,m}$

Fig. 3: The ovearll structure of GERA. The raw point cloud coordinate information is processed by an offline geometric constructor, resulting in encoded geometric information. This encoded data is passed through an MLP-based encoder, extracting $\mathcal{F}^{geo}_{\mathbf{P}_{\mathcal{S}}}$ and $\mathcal{F}^{geo}_{\mathbf{P}_{\mathcal{T}}}$. Subsequently, the original coordinates are concatenated with the geometric information, yielding a fully populated feature map $\mathcal{F}^{geo}_{\mathbf{full}} \in \mathbb{R}^{d\times n}$. Finally, the $\mathcal{F}^{geo}_{\mathbf{full}}$ is fed into a decoder, to obtain the displacement matrix.

lower than the corresponding values of 0.9968, 1.1702, and 1.2527 for position encoding. This demonstrates that our method can achieve robust and reliable feature extraction across different samples.

B. GERA architecture

1) Geometric Encoder and Decoder: As we discussed, offline geometric encoding extrated rich features from the raw point clouds. By leveraging these features, we can effectively and efficiently address the non-rigid registration task with a simple MLP architecture. Given the raw point cloud inputs $\mathbf{P}_{\mathcal{S}}$ and $\mathbf{P}_{\mathcal{T}}$, the geometric extraction process $Geo(\cdot)$ first constructs the geometric information. The resulting geometric embeddings from $\mathbf{P}_{\mathcal{S}}$ and $\mathbf{P}_{\mathcal{T}}$ can be expressed as:

$$\begin{aligned}\mathbf{P}^{geo}_{\mathcal{S}} &= Geo(\mathbf{P}_{\mathcal{S}}) = \left\{\mathbf{sc}^{i}_{\mathbf{P}_{\mathcal{S}}} \in \mathbb{R}^{d} \mid 1 \leq i \leq m\right\},\\ \mathbf{P}^{geo}_{\mathcal{T}} &= Geo(\mathbf{P}_{\mathcal{T}}) = \left\{\mathbf{sc}^{i}_{\mathbf{P}_{\mathcal{T}}} \in \mathbb{R}^{d} \mid 1 \leq i \leq m\right\},\end{aligned} \tag{7}$$

where $\mathbf{sc}$ denotes the edge length in the encoded fully connected graph, i, k are the point indices, and m represents the number of points in the point cloud.

The feature extraction of $\mathbf{P}_{\mathcal{S}}$ and $\mathbf{P}_{\mathcal{T}}$ is performed through an MLP, resulting in geometric features $\mathcal{F}^{geo}_{\mathbf{P}_{\mathcal{S}}}$ and $\mathcal{F}^{geo}_{\mathbf{P}_{\mathcal{T}}}$, respectively. These two features are then concatenated to generate a new representation concat$(\mathcal{F}^{geo}_{\mathbf{P}_{\mathcal{S}}}, \mathcal{F}^{geo}_{\mathbf{P}_{\mathcal{T}}})$. This process can be formulated as:

$$\mathcal{F}^{geo} = \text{concat}\left\{\mathbf{P}_{\mathcal{S}}, \lambda(\text{concat}(\mathcal{F}^{geo}_{\mathbf{P}_{\mathcal{S}}}, \mathcal{F}^{geo}_{\mathbf{P}_{\mathcal{T}}}), n), \mathbf{P}_{\mathcal{T}}\right\}, \tag{8}$$

where $\lambda(\rho, n)$ represents the replication of ρ n times.

The concatenated feature vector $\mathcal{F}^{geo} \in \mathbb{R}^{d}$ is replicated n times, where n is the number of points in the point cloud, producing a fully populated feature map $\mathcal{F}^{geo}_{\mathbf{full}} \in \mathbb{R}^{d\times n}$. This feature map concatenates in the packing stage with the point cloud coordinates. The decoder, an MLP, then processes this combined data to output the predicted displacement matrix for point cloud registration.

2) Loss function: Previous methods used 3D coordinates as a metric to quantify the differences between two point clouds. However, we found that the geometric information constructed by GERA can capture detailed geometric differences in the point clouds. Therefore, we introduced $\mathcal{L}_{geo}$, an unsupervised loss function that can be combined with existing registration methods, leading to significant improvements.

Our $\mathcal{L}_{geo}$ is formed by extracting geometric information from the target and output point clouds using the geometric extraction process $Geo(\cdot)$, with the error calculated via an adapted Chamfer distance loss.

$$\begin{aligned}\mathcal{L}_{geo} = &\sum_{\mathbf{p}\in\mathbf{P}_{\mathcal{S}}} \min_{\mathbf{q}\in\mathbf{P}_{\mathcal{T}}} \|Geo(\mathbf{p}) - Geo(\mathbf{q})\|^{2}\\ &+ \sum_{\mathbf{q}\in\mathbf{P}_{\mathcal{T}}} \min_{\mathbf{p}\in\mathbf{P}_{\mathcal{S}}} \|Geo(\mathbf{q}) - Geo(\mathbf{p})\|^{2},\end{aligned} \tag{9}$$

where $\mathbf{p}$ and $\mathbf{q}$ denote points in the source point cloud and the target point cloud.

Additionally, we employed the commonly used root mean square error (RMSE) to calculate the error between the source and target point clouds, defining it as $\mathcal{L}_{xyz}$. During training, $\mathcal{L}_{geo}$ and $\mathcal{L}_{xyz}$ form a semi-supervised combination learning framework, and the total loss can be expressed as:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{geo} + (1-\alpha)\mathcal{L}_{xyz}, \tag{10}$$

where $\alpha = 0$ corresponds to our proposed GERA-xyz, while $\alpha \neq 0$ corresponds to GERA-geo.

IV. Experiments and Results

A. Dataset

To simulate realistic surgical scenarios, we selected the point cloud subsets of the liver, brain, stomach, and small bowel from the MedShapeNet dataset [34]. The original point clouds, consisting of 10,000 points, were downsampled to 1,024 points to mimic the sparse point clouds commonly encountered in real-world scenarios. Two benchmarks are

TABLE I: Comparison of Various Methods on Liver, Brain and Stomach Point Cloud Datasets. The best results are in **bold** while the second best are <u>underlined</u>.

Method	Training	Years	RMSE (mm)	CD (mm)	TT (s)	IT (ms)	FLOPs (G)	Parameters (M)
BCPD [22]	-	2020	17.75	11.17	-	5105.63	-	-
PointSetReg [24]	-	2024	**7.07**	**2.71**	-	718.46	-	-
FPT [6]	Sup.	2021	12.80	8.08	30.48	8.23	7.58	5.61
PointPWC [25]	Sup.	2020	10.01	6.41	71.97	26.12	8.91	7.72
DifFlow3D [27]	Sup.	2024	13.51	8.15	155.26	53.39	16.97	3.51
Bi-pointflownet [26]	Sup.	2022	8.07	5.51	72.06	29.87	8.16	7.96
Lepard [7]	Sup.	2022	8.10	6.02	364.98	109.38	40.62	23.72
GERA-xyz (Ours)	Sup.	-	8.13	7.88	**26.10**	**6.16**	**3.19**	**2.96**
GERA-geo (Ours)	Semi.	-	<u>7.01</u>	<u>5.48</u>	<u>26.34</u>	**6.16**	**3.19**	**2.96**

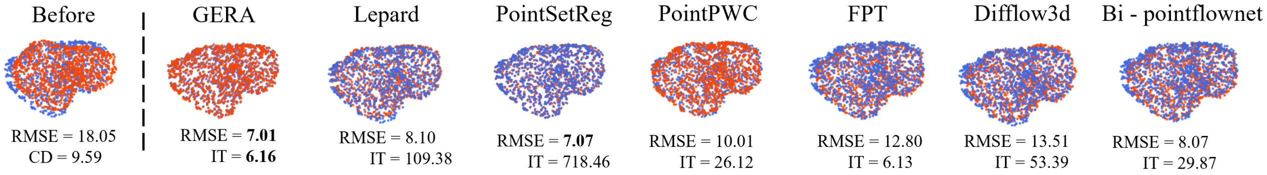

Fig. 4: The qualitative registration results on the dataset where the two point sets exhibit pure deformation. The blue and red point sets represent the source and target point clouds, respectively. For both Case 1 and Case 2, the maximum noise magnitude is 2 mm, and the deformation magnitude is 18 mm.

constructed, where the first contains 1,033 point clouds of livers, brains, and stomachs, while the other consists of 131 small bowel samples. The latter poses a significant challenge because of its large variability and the limited number of samples. For both benchmarks, the samples were divided into training, validation, and test sets with a ratio of 8: 1: 1. The intraoperative source point sets were generated using the Thin Plate Spline (TPS) [35] method, with a deformation value set at 19 mm to meet the requirements for organ registration. To simulate realistic scenarios, random noise was introduced ranging from 1 to 3 mm.

B. Evaluation metrics

To evaluate the registration quality, we utilize two different evaluation metrics, i.e., RMSE and Chamfer distance(CD) [6], [36]. In addition to quality, we also evaluate the efficiency through various metrics. Training time (TT) represents the time required for each model to train a single epoch. In practical applications, when encountering unseen classes, retraining may be necessary. Therefore, shorter training times are advantageous for the real-world deployment of models. Inference time(IT) refers to the time required by a trained model to process a single-point cloud during testing. To achieve real-time performance, the inference time should be minimized as much as possible. In addition, FLOPs and the parameter numbers are evaluated to measure the complexity and computational efficiency of the model.

C. Implementation Details

All methods were implemented using the PyTorch framework on a single GPU (Nvidia GeForce RTX 4090, 24GB). The model was further fine-tuned on 1024 points randomly sampled from the original point sets, each consisting of 10,000 points. The Adam optimizer was employed, with a batch size set to 1, and the network was trained for a total of 300 epochs. For the comparative methods, we utilized their publicly available code versions and setup for epochs, optimizers, and hyperparameters.

To ensure fairness, all comparative methods have been thoroughly retrained on our organ datasets.

D. Experiment Results

1) Comparisons with Other Methods on The Combined Dataset: Fig. 4 compares our method with other approaches. While FPT [6], PointPwc [25], Difflow3D [27], and Bi-pointflownet [26] have faster inference times, their registration errors of 12.80 mm, 10.01 mm, 13.51 mm, and 8.07 mm do not meet accuracy or real-time requirements. PointSetReg [24] achieves a lower error but processes only one frame per second, limiting its efficiency. In contrast, our method achieves the best result among learning-based methods with an error of 7.01 mm and real-time performance at 156 frames per second.

Table I shows quantitative results on the combined dataset of liver, brain, and stomach using RMSE. FPT yields a 13.4 mm error, while PointPWCNet [25] and Bi-pointflownet [26] achieve 10.01 mm and 8.07 mm, but their inference times remain too slow for real-time use. Lepard [7] reduces the error to 8.10 mm but also fails to meet real-time demands. PointSetReg [24] achieves a 7.07 mm error but requires 718.46 ms per point cloud. Our method, with a 7.01 mm

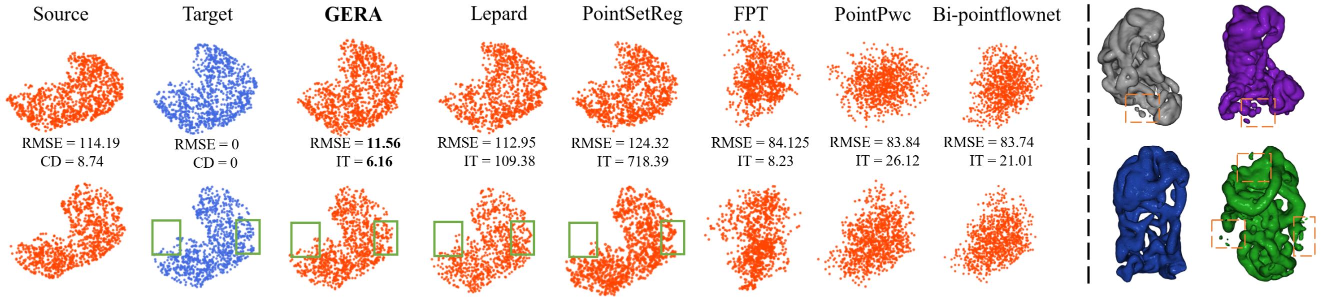

Fig. 5: The qualitative registration results on the dataset show that the two point sets exhibit pure deformation. The red and blue point sets represent the source and target point clouds, respectively. The deformation magnitude is 114.19 mm. On the right side of the figure, four randomly selected samples from the small intestine dataset are displayed, with yellow boxes marking breakpoints and noise locations.

error, performs real-time registration at 250 frames per second with the lowest FLOPs, at 20 percent of other methods. It also fine-tunes quickly, completing retraining in a fraction of the time needed by Bi-pointflownet [26] and Lepard [7], demonstrating superior efficiency and adaptability.

2) Results on Small Bowel Dataset: In robotic surgery, registration tests on multiple organs are often required. Additionally, point clouds collected from medical organs frequently contain substantial noise and are often incomplete in shape. To evaluate the robustness and capability of GERA, we conducted experiments on a more realistic and challenging small intestine dataset. These challenges arise from two main factors: first, the structure and distribution of the small intestine are more complex and diverse; second, the dataset is limited, containing only 131 samples, which accounts for 13% of the entire dataset. Fig. 5 qualitatively presents four small bowel samples with significant distribution differences, with noise and breakpoints highlighted in yellow boxes. Fig. 5 compares the performance of our method with other approaches. Scene flow methods such as FPT [6], PointPwc [25], and Bi-pointflownet [26] fail to perform effective point cloud registration, resulting in transformed point clouds that appear as scattered points. Although Lepard [7] and PointSetReg [24] maintain the shape of the small bowel point clouds, both methods tend to cluster points, leading to significantly higher RMSE values. In contrast, our method achieves the best registration results by aligning the transformed point cloud with the target point cloud while requiring the shortest inference time.

TABLE II: Benchmark Comparison of Various Methods. The best results are in **bold**, while the second best are underlined.

Method	**RMSE (mm)**	**CD (mm)**	**IT (ms)**	**TT(s)**
PointSetReg	124.32	**4.52**	718.39	-
Bi-pointflownet [26]	83.74	21.01	29.87	9.32
PointPWC [25]	83.84	20.92	26.12	9.14
Lepard [7]	112.95	9.03	109.38	193.12
FPT [6]	84.13	40.49	8.23	16.13
GERA-xyz	12.87	7.94	**6.16**	**3.02**
GERA-geo	**11.56**	7.74	**6.16**	3.09

Table II presents the quantitative results on the small bowel dataset using RMSE. We tested only those methods that performed well on the liver dataset. The FPT [6] method has an error of 84.13 mm, while PointPWCNet [25] and Bi-pointflownet [26] achieve errors of 83.84 mm and 83.74 mm, respectively. However, all three methods perform disastrously on the small intestine dataset and fail to complete the registration task. Although Lepard [7] and PointSetReg [24] show RMSE of 112.95 mm and 124.32 mm, respectively, due to their tendency to cluster points together, they maintain the basic shape of the point cloud and perform well in the CD metric, with values of 9.03 mm and 4.52 mm. Nonetheless, their inference times remain excessively long, severely impeding real-time point cloud registration. In contrast, our method achieves an error of 11.56 mm and is the only method capable of operating normally on the small intestine dataset, performing real-time registration at 250 frames per second. Additionally, it has significantly shorter fine-tuning and retraining times compared to other learning-based methods, demonstrating superior efficiency and adaptability.

V. CONCLUSION

In this paper, we introduced GERA, a method for real-time point cloud processing utilizing an offline geometric information constructor. We conducted experiments using MMD and a challenging small bowel dataset to validate the robustness and superiority of our approach. However, our experiments were primarily focused on the registration of individual organ data and did not consider the registration of composite objects in scenarios like scene flow. Our future research direction is to apply GERA's offline geometric encoder to the scene flow problem, aiming to achieve high-precision scene flow estimation with no time delay.

References

[1] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 529–537.

[2] Y. Shen, L. Hui, J. Xie, and J. Yang, "Self-supervised 3d scene flow estimation guided by superpoints," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5271–5280.

[3] J. Chang, Y. Xu, Y. Li, Y. Chen, and X. Han, "Gaussreg: Fast 3d registration with gaussian splatting," *arXiv preprint arXiv:2407.05254*, 2024.

[4] D. Cattaneo, M. Vaghi, and A. Valada, "Lcdnet: Deep loop closure detection and point cloud registration for lidar slam," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2074–2093, 2022.

[5] H. Lim, B. Kim, D. Kim, E. Mason Lee, and H. Myung, "Quatro++: Robust global registration exploiting ground segmentation for loop closing in lidar slam," *The International Journal of Robotics Research*, vol. 43, no. 5, pp. 685–715, 2024.

[6] Z. M. Baum, Y. Hu, and D. C. Barratt, "Real-time multimodal image registration with partial intraoperative point-set data," *Medical image analysis*, vol. 74, p. 102231, 2021.

[7] Y. Li and T. Harada, "Lepard: Learning partial point cloud matching in rigid and deformable scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5554–5564.

[8] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 143–11 152.

[9] R. H. Taylor, N. Simaan, A. Menciassi, and G.-Z. Yang, "Surgical robotics and computer-integrated interventional medicine [scanning the issue]," *Proceedings of the IEEE*, vol. 110, no. 7, pp. 823–834, 2022.

[10] M. J. Ringel, J. S. Heiselman, W. L. Richey, I. M. Meszoely, W. R. Jarnagin, and M. I. Miga, "Comparing regularized kelvinlet functions and the finite element method for registration of medical images to sparse organ data," *Medical Image Analysis*, p. 103221, 2024.

[11] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6411–6420.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[14] J. Serafin and G. Grisetti, "Nicp: Dense normal based point cloud registration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 742–749.

[15] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI Press, 2015, pp. 258–267.

[16] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[17] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[18] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[19] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[20] V. Panaganti and R. Aravind, "Robust nonrigid point set registration using graph-laplacian regularization," in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 1137–1144.

[21] G. Wang, Q. Zhou, and Y. Chen, "Robust non-rigid point set registration using spatially constrained gaussian fields," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1759–1769, 2017.

[22] O. Hirose, "A bayesian formulation of coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 7, pp. 2269–2286, 2020.

[23] J. Ma, J. Zhao, J. Jiang, and H. Zhou, "Non-rigid point set registration with robust transformation estimation under manifold regularization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[24] M. Zhao, J. Jiang, L. Ma, S. Xin, G. Meng, and D.-M. Yan, "Correspondence-free non-rigid point set registration using unsupervised clustering analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 199–21 208.

[25] W. Wu, Z. Y. Wang, Z. Li, W. Liu, and L. Fuxin, "Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 88–107.

[26] W. Cheng and J. H. Ko, "Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation," in *European Conference on Computer Vision*. Springer, 2022, pp. 108–124.

[27] J. Liu, G. Wang, W. Ye, C. Jiang, J. Han, Z. Liu, G. Zhang, D. Du, and H. Wang, "Difflow3d: Toward robust uncertainty-aware scene flow estimation with iterative diffusion-based refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 109–15 119.

[28] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, *Distance geometry: theory, methods, and applications*. Springer Science & Business Media, 2012.

[29] A. Cassioli, B. Bardiaux, G. Bouvier, A. Mucherino, R. Alves, L. Liberti, M. Nilges, C. Lavor, and T. E. Malliavin, "An algorithm to enumerate all possible protein conformations verifying a set of distance constraints," *BMC bioinformatics*, vol. 16, pp. 1–15, 2015.

[30] C. Lavor, L. Liberti, B. Donald, B. Worley, B. Bardiaux, T. E. Malliavin, and M. Nilges, "Minimal nmr distance information for rigidity of protein graphs," *Discrete Applied Mathematics*, vol. 256, pp. 91–104, 2019.

[31] H. Cao, Y. Xu, J. Yang, P. Yin, S. Yuan, and L. Xie, "Mopa: Multi-modal prior aided domain adaptation for 3d semantic segmentation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9463–9470.

[32] L. Kong, N. Quader, and V. E. Liong, "Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9338–9345.

[33] M. Rochan, S. Aich, E. R. Corral-Soto, A. Nabatchian, and B. Liu, "Unsupervised domain adaptation in lidar semantic segmentation with self-supervision and gated adapters," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2649–2655.

[34] J. Li, A. Pepe, C. Gsaxner, G. Luijten, Y. Jin, N. Ambigapathy, E. Nasca, N. Solak, G. M. Melito, A. R. Memon *et al.*, "Medshapenet: A large-scale dataset of 3d medical shapes for computer vision," *arXiv preprint arXiv:2308.16139*, 2023.

[35] F. L. Bookstein and W. Jaynes, "Thin-plate splines and the analysis of biological shape," *Video tape*, vol. 21, 1990.

[36] T. Nguyen, Q.-H. Pham, T. Le, T. Pham, N. Ho, and B.-S. Hua, "Point-set distances for learning representations of 3d point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 478–10 487.