

# Deep-PE: A Learning-Based Pose Evaluator for Point Cloud Registration

Junjie Gao<sup>1</sup>, Chongjian Wang<sup>2</sup>, Zhongjun Ding<sup>3</sup>, Shuangmin Chen<sup>4</sup>, Shiqing Xin<sup>1</sup>, Changhe Tu<sup>1</sup>, Wenping Wang<sup>5</sup>,

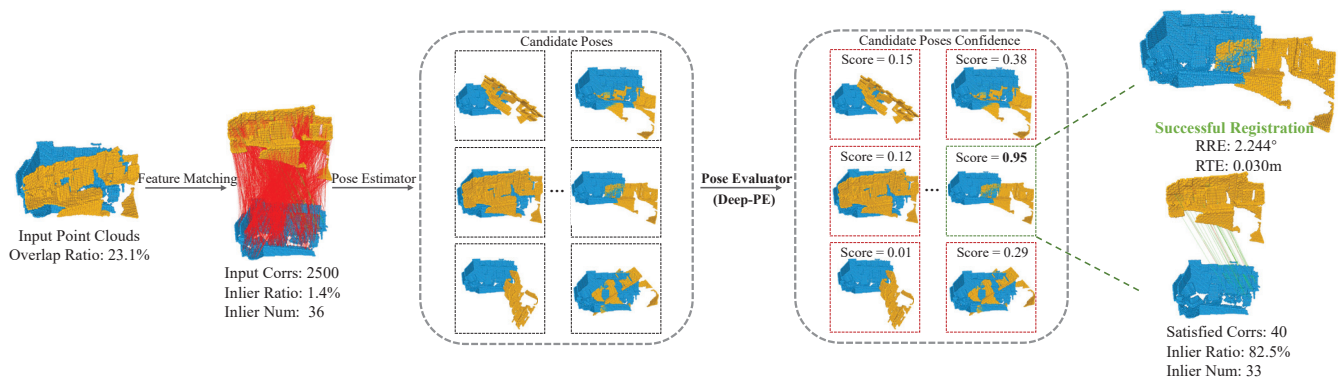
<sup>1</sup>Shandong University

<sup>2</sup>Shandong University of Science and Technology

<sup>3</sup>National Deep Sea Center

<sup>4</sup>Qingdao University of Science and Technology

<sup>5</sup>Texas A&M University



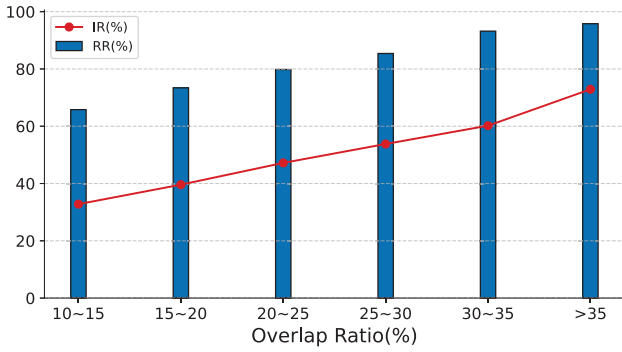
**Figure 1:** Visualization of Deep-PE in the estimator-based registration pipeline. Such pipelines typically involve establishing correspondences between point clouds using feature descriptors and generating a set of candidate poses using a pose estimator. Subsequently, our proposed Deep-PE predicts the confidence for each candidate pose and selects the pose with the highest confidence score as the final transformation. In this specific example, our approach is capable of accurately determining the correct transformation even if the inlier ratio of input correspondences is as low as 1.4%. **RRE:** Relative Rotation Error. **RTE:** Relative Translation Error. **# Input Corrs:** No. of input correspondences. **# Satisfied Corrs:** No. of correspondences consistent with the predicted pose.

## Abstract

In the realm of point cloud registration, the most prevalent pose evaluation approaches are statistics-based, identifying the optimal transformation by maximizing the number of consistent correspondences. However, registration recall decreases significantly when point clouds exhibit a low overlap ratio, despite efforts in designing feature descriptors and establishing correspondences. In this paper, we introduce Deep-PE, a lightweight, learning-based pose evaluator designed to enhance the accuracy of pose selection, especially in challenging point cloud scenarios with low overlap. Our network incorporates a Pose-Aware Attention (PAA) module to simulate and learn the alignment status of point clouds under various candidate poses, alongside a Pose Confidence Prediction (PCP) module that predicts the likelihood of successful registration. These two modules facilitate the learning of both local and global alignment priors. Extensive tests across multiple benchmarks confirm the effectiveness of Deep-PE. Notably, on 3DLoMatch with a low overlap ratio, Deep-PE significantly outperforms state-of-the-art methods by at least 8% and 11% in registration recall under handcrafted FPFH and learning-based FCGF descriptors, respectively. To the best of our knowledge, this is the first study to utilize deep learning to select the optimal pose without the explicit need for input correspondences.

## CCS Concepts

• Computing methodologies → Matching;



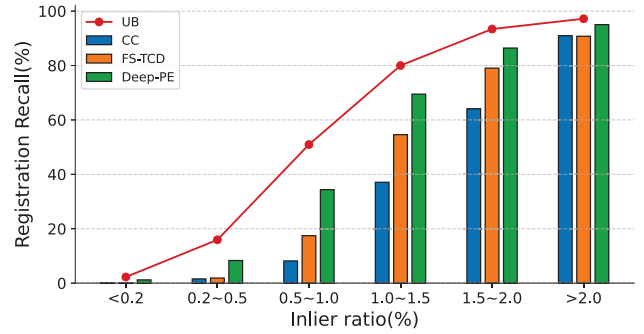
**Figure 2:** Even in the well-known registration pipeline [QYW\*22], the Inlier Ratio (IR) and Registration Recall (RR) deteriorates rapidly for point cloud pairs with an overlap of <30%.

## 1. Introduction

Finding an accurate rigid transformation between two unaligned partial point clouds—a process known as point cloud registration—is a fundamental task in the fields of graphics, vision, and robotics. This task becomes increasingly challenging with low-overlap point clouds. In response to this, a significant number of advanced feature descriptors [HGU\*21; YLS\*21; QYW\*22] and robust pose estimators [CSYT22; JDW\*23; ZYZZ23] have been developed. However, recent progress in registration seems to have encountered a bottleneck, especially in improving the registration recall in low-overlap scenes.

As shown in Figure 2, we find that even in the well-known registration pipeline [QYW\*22], when the overlap ratio drops below 30%, the inlier ratio (IR) of extracted correspondences rapidly decreases, leading to a simultaneous decrease in registration recall (RR). To explore this phenomenon further, we extract correspondences using the PPFH descriptor [RBB09] and generate candidate poses using SC2-PCR [CSYT22] on the low-overlap 3DLoMatch dataset. Subsequently, based on the inlier ratio, we divide all point cloud pairs into six different groups. As illustrated in Figure 3, it can be seen that although the pose estimator can generate a set of candidate poses containing the correct transformation, this represents the upper bound (UB) of the pose evaluator’s performance. However, during the process of selecting the optimal pose, the most commonly used and advanced statistics-based pose evaluators, CC [FB81] and FS-TCD [CSY\*23], exhibit varying degrees of performance degradation when the inlier ratio falls below 1%, largely due to their excessive reliance on the quality of input correspondences.

Motivated by this observation, we propose a lightweight deep learning-based pose evaluator named Deep-PE to enhance the accuracy of pose selection. Unlike traditional pose evaluators, Deep-PE fully integrates alignment priors, endowing it with global pose perception capabilities and making it sensitive solely to pose without dependence on the quality of input correspondences. Consequently, as demonstrated in Figure 3, our method shows stronger robustness against low inlier ratios. Additionally, unlike existing pose evaluators [FB81; YHQ\*21; CSY\*23] that must select one of the candidate poses as the final transformation, Deep-PE predicts



**Figure 3:** In low-overlap 3DLoMatch benchmark, compared to the statistics-based pose evaluators, CC [FB81] and FS-TCD [CSY\*23], Deep-PE demonstrates a significant advantage in cases of low inlier ratios and also approaches the upper bound (UB) value as the inlier ratio increases.

with a confidence score, enabling it to identify registration failures effectively.

To summarize, the main contributions are as follows:

- A pioneering learning-based pose evaluator is proposed, which does not rely on the quality of input correspondences and has the capability to rectify registration failures.
- A pose-aware attention module is proposed to simulate and learn the alignment status of point clouds under various candidate poses, and a pose confidence prediction module is proposed to predict the likelihood of successful registration.
- A weighted cross-entropy loss is proposed, which reweights the loss of each candidate pose according to its similarity with the ground truth for better identification of the optimal transformation.
- A series of experimental results demonstrate that our proposed method performs favorably against state-of-the-art pose evaluators.

## 2. Related Work

### 2.1. Estimator-based Registration

The registration pipeline of such methods typically includes three steps: firstly, feature descriptors [TSdS10; RBB09; GZWW18; CPK19; AHY\*20; WLH\*23; YLS\*21; QYW\*22; BLZ\*20; HGU\*21; LWZ\*19] are meticulously designed to describe the input point clouds for extracting correspondences; secondly, robust pose estimators [FB81; BM17; LH05; CDK20; BLZ\*21; PMR\*19; CSY\*23; ZYZZ23; JDW\*23] generate a series of candidate poses; finally, an effective pose evaluator [FB81; YHQ\*21; CSY\*23] is used to select the optimal pose from them as the final transformation.

For the methods involved in the first step, they can be divided into two categories based on how they extract correspondences. The first category aims to detect more repeatable keypoints [BLZ\*20; HGU\*21] and learn more powerful keypoint descriptors [LWZ\*19]. The second category considers all possible correspondences without keypoint detection [TSdS10; RBB09; GZWW18;

CPK19; AHY\*20; WLH\*23; YLS\*21; QYW\*22]. When extracting correspondences between two point clouds, points located outside the overlapping region can be entirely regarded as interference factors. This leads to the inability of these two categories of methods to achieve satisfactory performance on low-overlap point clouds, despite their excellent performance in high-overlap scenarios.

For the pose estimator mentioned in the second step, significant progress has been made in recent years, and it can be broadly categorized into two categories: traditional [FB81; BM17; LH05; CSY\*23; ZYZZ23] and learning-based [CDK20; BLZ\*21; PMR\*19]. Traditional pose estimators typically obtain outlier-free subsets for pose estimation through extensive iterations [FB81; BM17; MTN\*02] or checking compatibility [LH05; CSYT22] between correspondences. Learning-based pose estimators [CDK20; PMR\*19] typically consist of a classification network to reject outliers and a regression network to compute transformations.

As the final step in the registration pipeline, the pose evaluation has been greatly neglected, to the best of our knowledge, all current pose evaluators [FB81; YHQ\*21; CSY\*23] are statistics-based, assessing the alignment quality of candidate poses by maximizing the number of consistent correspondences. In high-overlap scenarios, a high inlier ratio typically supports this evaluation mechanism. However, low-overlap point clouds often come with a much lower inlier ratio, rendering this reliance on the quality of input correspondences ineffective. In contrast, our proposed method fully learns the alignment status of point clouds under various candidate poses and predicts the likelihood of successful registration in terms of confidence. Compared to traditional statistics-based pose evaluators, our approach offers higher selecting precision and the ability to identify registration failures.

## 2.2. Estimator-free Registration

Some efforts have diverged from the traditional approach of combining learned descriptors with robust pose estimators, choosing instead to fully integrate the pose estimation process within their training pipeline [WS19; YEK\*20; LPL21; HMZ20; YL20; MTH\*23]. These methods can be further classified into three categories. The first category [WS19; YL20] follows the ICP approach [BM92], iteratively establishing soft correspondences and computing transformations using a differentiable weighted SVD [PL00]. The second category [LPL21; HMZ20] first extracts a global feature vector [QSMG16] for each point cloud and then regresses transformations using these feature vectors. The final category is distribution-level methods [YEK\*20; MTH\*23], which model point clouds as probability distributions (e.g., Gaussian mixture models [JV10]) and perform alignment using correlation-based optimization frameworks. Although estimator-free registration pipelines avoid some issues introduced by pose estimators and achieve promising results on single synthetic shapes, they can fail in large-scale scenes and are unable to handle more challenging scenarios such as low-overlap point clouds. Combining advanced feature descriptors with a robust pose estimator is sufficient to generate a set of candidate poses containing the correct transformation. Therefore, in the network design, we fully leverage the advantages of an estimator-free registration pipeline, opting for the network

to select the optimal ones from candidate poses instead of directly regressing pose parameters. This strategy significantly reduces the training difficulty of the network.

## 2.3. Transformer in Vision

Convolutional Neural Networks (CNNs) excel in local feature description, but are limited by their receptive field size, whereas Transformer networks [VSP\*17] are better at capturing long-range contextual information and facilitating effective information exchange between inputs. Therefore, these two methods complement each other and are widely applied in 2D and 3D feature matching tasks [WPN\*19; SSW\*21; HGU\*21; YLS\*21; QYW\*22]. However, Transformers are computationally expensive and typically can only be used at low-resolution levels. With the development of deep learning, numerous neural network frameworks [QSMG16; TQD\*19; CPK19; WSL\*19] have emerged, but enabling the network to learn the alignment status under different poses is crucial for our task. Recently, we find that the Epipolar Transformer [HYFY20] with constrained attention regions has achieved remarkable success in 2D stereo matching tasks. Inspired by this, we propose a lightweight pose-aware attention module. The module can flexibly adjust attention regions based on different candidate poses, and it is local and limited. Therefore, on the one hand, it can simulate and learn the alignment status of point clouds under various candidate poses, while on the other hand, it significantly reduces computational costs. Then, by combining it with the pose confidence prediction module, we can predict the likelihood of successful registration for each candidate pose.

## 3. Method

### 3.1. Insight

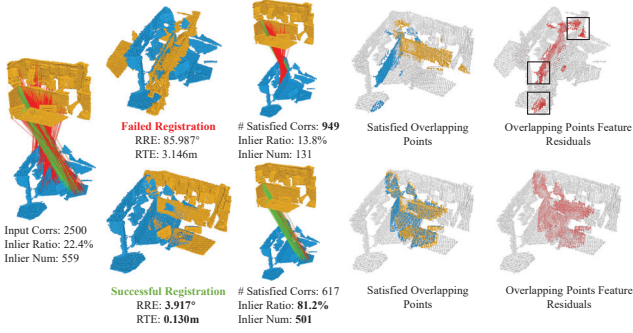
Our research builds on the following observation: As depicted in Figure 4, when the inlier ratio of input correspondences is low, overly relying on maximizing consistent correspondences for pose estimation is often ineffective. Incorrect poses result in large feature residuals within overlapping regions (see highlighted windows).

### 3.2. Overview

Given two point clouds:  $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}$  and  $\mathcal{Q} = \{\mathbf{q}_j \in \mathbb{R}^3 \mid j = 1, \dots, M\}$ , and a set of candidate poses  $\mathcal{H} \in \{\mathbf{R}, \mathbf{t}\}$  generated by the pose estimator, Deep-PE processes each candidate pose  $\mathcal{H}_i$  and predicts the relevant confidence score  $\mathcal{S}_i$ , which describes the alignment quality of the input point clouds. As shown in Figure 5, Deep-PE mainly consists of three components: a Feature Extraction (FE) module, a Pose-Aware Attention (PAA) module, and a Pose Confidence Prediction (PCP) module. The last two modules are our main design.

### 3.3. Feature Extractor

Due to the typically large number of candidate poses generated by pose estimators, a lightweight network design is essential to ensure efficient batch processing. Additionally, our primary focus is on the training aspects of pose evaluation rather than feature extraction. To



**Figure 4:** Illustration of our insight: Incorrect poses frequently align areas in the point clouds that should not be aligned; points within these regions typically display larger feature residuals. The fourth column depicts the overlapping regions predicted under the ground truth pose, while the fifth column illustrates the severity of feature residuals, with red colors denoting larger residuals. RRE: Relative Rotation Error. RTE: Relative Translation Error.

reduce training complexity and improve model convergence speed, we choose to utilize the pre-trained Geotransformer [QYW\*22] as the feature extractor, which employs KPConv [TQD\*19] to down-sample the input point cloud and simultaneously extract point-wise features. As depicted in Figure 5, we only utilize two layers of points and their corresponding features. To further enhance feature learning and facilitate subsequent computations, we employ two linear layers to map these features to the same dimension. Specifically, the points at the coarsest layers are represented as  $\hat{\mathcal{P}}$  and  $\hat{\mathcal{Q}}$ , with their corresponding feature matrices  $\hat{\mathbf{F}}^{\mathcal{P}} \in \mathbb{R}^{|\hat{\mathcal{P}}| \times d}$  and  $\hat{\mathbf{F}}^{\mathcal{Q}} \in \mathbb{R}^{|\hat{\mathcal{Q}}| \times d}$ . At the penultimate layers, the points are represented as  $\tilde{\mathcal{P}}$  and  $\tilde{\mathcal{Q}}$ , with their corresponding feature matrices  $\tilde{\mathbf{F}}^{\mathcal{P}} \in \mathbb{R}^{|\tilde{\mathcal{P}}| \times d}$  and  $\tilde{\mathbf{F}}^{\mathcal{Q}} \in \mathbb{R}^{|\tilde{\mathcal{Q}}| \times d}$ .

### 3.4. Pose-Aware Attention Module

Given that Deep-PE needs to predict the confidence of each candidate pose, we choose to cache and reuse the obtained features. This design not only reduces computational costs but also avoids potential adverse effects on training caused by feature divergence resulting from each prediction.

To enable the network to learn and evaluate the alignment quality of different candidate poses, we simulate the alignment status of point clouds by adjusting the attention region. Specifically, under the current candidate pose  $\mathcal{H}_i$ , each point  $\hat{\mathbf{p}}_i$  in  $\hat{\mathcal{P}}$  with associated feature  $\hat{\mathbf{f}}_i^{\mathcal{P}}$  has a corresponding point defined as  $\hat{\mathbf{p}}'_i = \mathcal{H}_i \hat{\mathbf{p}}_i$ , and then we adopt a  $k$ -nearest neighbor search within  $\hat{\mathcal{Q}}$  to produce  $k$  points  $\mathcal{K}_{\hat{\mathbf{p}}_i}^{\hat{\mathcal{Q}}}$ . The associated learned features are denoted as  $\mathcal{F}_{\hat{\mathbf{p}}_i}^{\hat{\mathcal{Q}}} \in \mathbb{R}^{k \times d}$ . If the distance from the sampling points to  $\hat{\mathbf{p}}'_i$  exceeds the threshold  $t$ , we zero-pad the features. Thus, we build a feature volume  $\mathcal{F}^{\hat{\mathcal{P}} \rightarrow \hat{\mathcal{Q}}} \in \mathbb{R}^{|\hat{\mathcal{P}}| \times k \times d}$ . The feature volume  $\mathcal{F}^{\hat{\mathcal{Q}} \rightarrow \hat{\mathcal{P}}}$  for  $\hat{\mathcal{Q}}$  is computed and denoted similarly.

Accordingly, given the coarse point features  $\hat{\mathbf{F}}^{\mathcal{P}}$  and the relevant feature volume  $\mathcal{F}^{\hat{\mathcal{P}} \rightarrow \hat{\mathcal{Q}}}$ , the pose-aware attention feature

$\hat{\mathbf{H}}^{\mathcal{P}} \in \mathbb{R}^{|\hat{\mathcal{P}}| \times d}$  is the weighted sum of all projected feature volumes:

$$\hat{\mathbf{h}}_i^{\mathcal{P}} = \sum_{j=1}^k a_{i,j} (\mathbf{f}_j \mathbf{W}^V) \quad (1)$$

where  $\mathbf{f}_j \in \mathcal{F}^{\hat{\mathcal{P}} \rightarrow \hat{\mathcal{Q}}}$ , and the weight coefficient  $a_{i,j}$  is computed by a row-wise softmax on the attention score  $e_{i,j}$ , with  $e_{i,j}$  computed as:

$$e_{i,j} = \frac{(\mathbf{f}_i \mathbf{W}^Q) (\mathbf{f}_j \mathbf{W}^K)^T}{\sqrt{d}} \quad (2)$$

Here,  $\mathbf{f}_i \in \hat{\mathbf{F}}^{\mathcal{P}}$ , and  $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$  are the respective projection matrices for queries, keys, and values. The pose-aware attention features  $\hat{\mathbf{H}}^{\mathcal{Q}}$  for  $\hat{\mathcal{P}}^{\mathcal{Q}}$  are computed in the same manner.

### 3.5. Pose Confidence Prediction Module

After passing through the pose-aware attention module, the coarse point features  $\hat{\mathbf{F}}^{\mathcal{P}}$  and  $\hat{\mathbf{F}}^{\mathcal{Q}}$  are updated to  $\hat{\mathbf{H}}^{\mathcal{P}}$  and  $\hat{\mathbf{H}}^{\mathcal{Q}}$ . As illustrated in Figure 6, notable disparities between the features updated with correct and incorrect poses are observed. The primary reason lies in the attention mechanism's ability to assign greater weight to similar features. In this scenario, when the pose is correct, the corresponding nearest neighbor features tend to exhibit some similarities, and vice versa.

As depicted in Figure 5, to discern the variance in feature updates across different poses, we implement a residual operation on the acquired features. Specifically,  $\hat{\mathbf{R}}^{\mathcal{P}} = \hat{\mathbf{H}}^{\mathcal{P}} - \hat{\mathbf{F}}^{\mathcal{P}}$ , with  $\hat{\mathbf{R}}^{\mathcal{Q}}$  calculated similarly. Drawing inspiration from the estimator-free registration pipeline [LPL21] and point cloud classification research [QSMG16], we concatenate the resulting residual features:  $\text{Cat}(\hat{\mathbf{R}}^{\mathcal{P}}, \hat{\mathbf{R}}^{\mathcal{Q}}) \in \mathbb{R}^{|\mathcal{P}+\mathcal{Q}| \times d}$  and then conduct a max-pooling operation to derive the global feature:  $\mathbf{G} \in \mathbb{R}^{1 \times d}$ . This diversity furnishes a pivotal cue for the network to grasp and assess the alignment quality of various candidate poses. Subsequently, we employ a three-layer Multilayer Perceptron (MLP) to map and acquire the corresponding confidence scores. After the initial two linear layers, each layer is followed by 1D batch normalization, a leaky ReLU, and a dropout layer. Given that we frame the problem as binary classification, we apply the sigmoid layer subsequent to the final linear layer. The detailed network structure is outlined in the Appendix.

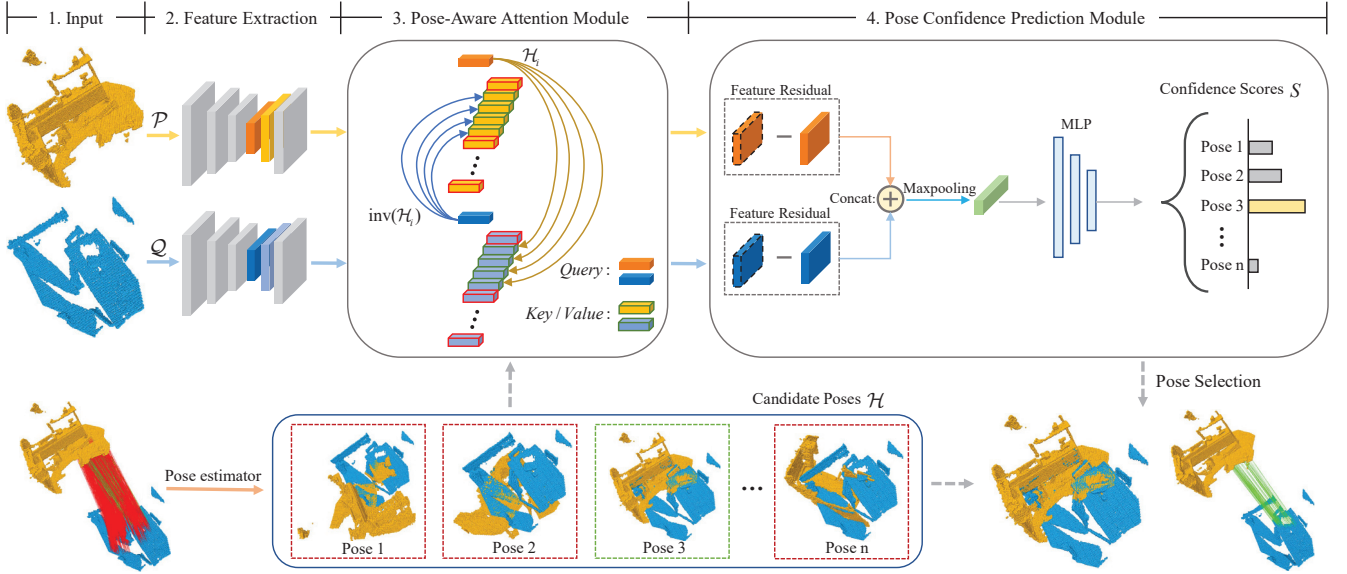
### 3.6. Loss Function

Distinguishing between correct and incorrect poses can be framed as a binary classification task, thus, we can utilize cross-entropy loss to train the network:

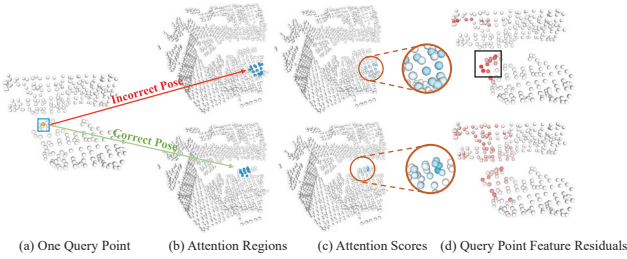
$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \mathcal{S}_i \log(\mathcal{S}_i) + (1 - \mathcal{S}_i) \log(1 - \mathcal{S}_i) \quad (3)$$

where  $\mathcal{S}_i$  denotes the predicted confidence score for each candidate pose  $\mathcal{H}_i$  in the training batch  $\mathcal{B}$ , and  $\mathcal{S}_i$  represents the ground-truth label, which is either 1 or 0, indicating whether the candidate pose  $\mathcal{H}_i$  is correct or incorrect.

However, directly employing Eq 3 would treat each candidate



**Figure 5:** *Deep-PE* mainly consists of three components: 1. The pre-trained feature extractor downsamples the input point clouds and learns features in multiple resolution levels. The points and features of the coarsest and penultimate layers are cached and reused for each candidate pose  $\mathcal{H}_i$ . 2. The pose-aware attention module adjusts attention regions based on different candidate poses and embeds relevant features into coarse-level features. 3. The pose confidence prediction module first calculates the feature residuals before and after coarse-level feature updates for each point cloud, then combines them through concatenation and max-pooling operations. Finally, a Multilayer Perceptron (MLP) layer is employed to predict the confidence score  $\mathcal{S}_i$  associated with each candidate pose  $\mathcal{H}_i$ . Then, the pose with the maximum confidence score is selected as the final transformation.



**Figure 6:** *Illustration of attention regions and scores under correct and incorrect poses. It's evident that, with the correct pose, attention scores are higher in regions resembling the query point, leading to smaller feature residuals before and after updating. Conversely, the opposite holds true under an incorrect pose.*

pose equally. Our ultimate objective is to select the optimal transformation from the candidate poses, thereby making it more crucial to learn the evaluation capability of correct poses compared to that of incorrect poses with high errors. Since a distinction exists between the candidate pose  $\mathcal{H}_i$  and the ground truth pose, denoted as  $d(\mathcal{H}_i)$ , we incorporate a smoothing operation into the loss function:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \bar{\mathcal{S}}_i \cdot (\alpha \cdot (\beta - d(\mathcal{H}_i))) \cdot \log(\mathcal{S}_i) + (1 - \bar{\mathcal{S}}_i) \cdot (1 - \alpha \cdot (d(\mathcal{H}_i) - \beta))^\gamma \cdot \log(1 - \mathcal{S}_i) \quad (4)$$

where  $\alpha$  and  $\gamma$  are weighting parameters. Typically,  $\beta$  serves as a threshold used to determine whether the pose is correct or incorrect. When the ground truth label  $\bar{\mathcal{S}}_i = 1$ ,  $d(\mathcal{H}_i)$  is always less than  $\beta$ . A smaller  $d(\mathcal{H}_i)$  indicates a more accurate pose, hence,  $\beta - d(\mathcal{H}_i)$  exerts a greater impact on the loss. When  $d(\mathcal{H}_i)$  approaches the boundary, its influence on the loss diminishes, and vice versa. Here,  $d(\mathcal{H}_i)$  represents the Root Mean Square Error (RMSE) of the ground-truth correspondences under the pose  $\mathcal{H}_i$ .

## 4. Experiments

### 4.1. Datasets

We evaluate our model on three public benchmarks, including indoor, outdoor and multi-way registration scenes. Follow [HGU\*21], for indoor scenes, we evaluate our model on both 3DMatch, where point cloud pairs share  $> 30\%$  overlap, and 3DLoMatch, where point cloud pairs have  $10\% \sim 30\%$  overlap. In line with existing works [CSYT22; CSY\*23], we evaluate for outdoor scenes on odometry KITTI, where point cloud pairs share  $> 30\%$  overlap. In addition, following [CSY\*23], we also use augmentation ICL\_NUIM dataset for testing the performance on the multi-way registration task.

### 4.2. Experimental Setup

**Metrics.** For indoor and outdoor scenes, we adopt three typically used metrics, including Relative Rotation Error (RRE), Relative

Translation Error (RTE), and Registration Recall (RR). Specifically, RRE and RTE represent the rotation and translation errors between the estimated pose and the ground truth pose, respectively. The RR is defined as the fraction of the point cloud pairs whose RRE and RTE are both below certain thresholds (i.e., for indoor scenes  $RRE < 15^\circ$ ,  $RTE < 0.3m$ ; for outdoor scenes,  $RRE < 5^\circ$ ,  $RTE < 0.6m$ ). In addition, we propose a novel metric, Failure Scenes Recognition Recall (FSRR), to assess the capability of a pose evaluator in identifying the proportion of point cloud pairs in candidate poses that do not contain the correct transformation. For the multi-way registration scenes, following [CSY\*23; CDK20], we report the absolute trajectory error (ATE, cm) as the measurement. The detailed definitions of the metrics are provided in the Appendix.

**Implementation details.** To construct the training and validation sets, we utilized the Geotransformer [QYW\*22] to extract correspondences for each point cloud pair and employed the SC2-PCR [CSYT22] to generate sufficient candidate poses. During the training phase, we ensured that each batch contained 10 correct poses and 10 incorrect poses to maintain a balanced distribution of positive and negative samples. We implement and evaluate our Deep-PE with PyTorch on an AMD EPYC 7642 and an NVIDIA RTX 3090 GPU. The network is trained with the Adam optimizer for 40 epochs on 3DMatch and 80 epochs on KITTI. The batch size is 20, and the weight decay is  $1 \times 10^{-5}$ . The learning rate starts from  $1 \times 10^{-6}$  and decays exponentially by 0.05 every epoch on 3DMatch and every 4 epochs on KITTI. In the feature extraction module, the linear layer maps uniformly to dimension  $d = 256$ . In the pose-aware attention module, the number of nearest neighbors  $k$  is set to 16. For the parameters of the loss function,  $\alpha$ ,  $\beta$ , and  $\gamma$  are set to 5, 0.2, and 2, respectively, ensuring that the weight terms are within the range of 0 to 1. In the pose confidence prediction module, the parameter of the dropout layer is set to 0.5. To ensure a fair comparison of the performance of different evaluators, we used SC2-PCR [CSYT22] to generate 1,000 candidate poses under different descriptors as the test set, and then compared our method with all known evaluators. In the actual test, to improve the speed of pose evaluation of our model, CC [FB81] is used to preprocess the candidate poses  $\mathcal{H}$ , and  $|\mathcal{H}| * \delta$  poses are selected in descending order according to their numerical counts. Based on our experience, we have found that setting  $\delta = 0.4$  and approximately 400 candidate poses achieves a balance between performance and speed.

### 4.3. Evaluation on Indoor Scenes

**Comparison with Different Evaluators.** We compared our method with all known pose evaluators, including Correspondences Counting (CC) [FB81], Mean Average Error (MAE) [YHQ\*21], Mean Square Error (MSE) [YHQ\*21], and Feature and Spatial consistency constrained Truncated Chamfer Distance (FS-TCD) [CSY\*23]. It should be noted that SC2-PCR [CSYT22] is used here to produce candidate poses, and different pose evaluators are used to select the final transformation. As shown in Tab. 1, the improvements in MAE [YHQ\*21] and MSE [YHQ\*21] are very limited compared to the most commonly used CC [FB81], which is also reported in the latest work [ZYZZ23]. By establishing more correspondences and combining chamfer distance to an-

**Table 1:** Evaluation results with different pose evaluators on 3DMatch and 3DLoMatch.

	FPFH					
	3DMatch			3DLoMatch		
	RR (%) $\uparrow$	RRE (deg) $\downarrow$	RTE (cm) $\downarrow$	RR (%) $\uparrow$	RRE (deg) $\downarrow$	RTE (cm) $\downarrow$
CC [FB81]	83.98	2.24	6.80	36.83	4.16	10.23
MAE [YHQ*21]	83.92	2.18	6.76	37.84	3.95	10.08
MSE [YHQ*21]	83.30	2.14	6.74	37.11	3.86	9.94
FS-TCD [CSY*23]	87.18	2.19	6.81	42.45	4.01	10.55
Deep-PE (ours)	<b>89.40</b>	2.18	<b>6.52</b>	<b>50.87</b>	4.32	11.24
	FCGF					
	3DMatch			3DLoMatch		
	RR (%) $\uparrow$	RRE (deg) $\downarrow$	RTE (cm) $\downarrow$	RR (%) $\uparrow$	RRE (deg) $\downarrow$	RTE (cm) $\downarrow$
CC [FB81]	94.15	1.92	6.28	57.83	3.82	10.51
MAE [YHQ*21]	94.02	1.87	6.24	58.45	3.84	10.52
MSE [YHQ*21]	94.09	1.86	6.20	58.45	3.81	10.51
FS-TCD [CSY*23]	94.15	1.87	6.20	61.15	3.68	10.38
Deep-PE (ours)	<b>95.07</b>	1.87	6.24	<b>72.71</b>	3.97	10.89
	Geotransformer					
	3DMatch			3DLoMatch		
	RR (%) $\uparrow$	RRE (deg) $\downarrow$	RTE (cm) $\downarrow$	RR (%) $\uparrow$	RRE (deg) $\downarrow$	RTE (cm) $\downarrow$
CC [FB81]	95.69	1.92	5.75	77.99	3.01	8.66
MAE [YHQ*21]	95.63	1.92	5.73	78.33	2.97	8.60
MSE [YHQ*21]	95.69	1.92	5.75	78.44	2.96	8.62
FS-TCD [CSY*23]	95.75	1.92	5.67	78.72	2.95	8.47
Deep-PE (ours)	<b>96.24</b>	1.92	5.78	<b>80.24</b>	2.95	8.52

alyze the fitting quality of the point clouds, FS-TCD [CSY\*23] shows some improvements. However, the RR of our method is 2.22%/8.42% higher than that of statistics-based pose evaluators when using FPFH [RBB09], 0.92%/11.56% higher when using FCGF [CPK19], and 0.49%/1.49% higher when using Geotransformer [QYW\*22]. It's worth noting that RRE/RTE represent the mean errors of successfully registered point cloud pairs. As mentioned in [CSY\*23], this metric makes methods with high RR more likely to produce larger average errors because they include more challenging data when computing the average error. Nevertheless, Deep-PE still achieves competitive results on RRE and RTE.

In addition, we also present some qualitative results on the 3DLoMatch benchmark. As illustrated in Figure 7 and Figure 8, we use handcrafted FPFH [RBB09] and the advanced Geotransformer (GEO) [QYW\*22] extract correspondences respectively, we compared our method to the commonly used CC [FB81] and the advanced FS-TCD [CSY\*23]. We reported the visual registration and correspondence results in challenging scenarios. When the inlier ratio of input correspondences is low, statistics-based evaluation mechanisms become ineffective, leading to registration failures. In contrast, our method incorporates global alignment priors and utilizes a carefully designed neural network to learn and evaluate the quality of pose alignment. Consequently, it consistently identifies the optimal pose among the candidates, even if it does not satisfy the majority of correspondences.

**Combination with Different Estimators.** We compared our method with commonly used and advanced pose evaluators CC [FB81] and FS-TCD [CSY\*23], incorporating various pose estimators including SM [LH05], PointDSC [BLZ\*21], and SC2-PCR [CSYT22]. Similarly, each method was tested under FPFH [RBB09] and FCGF [CPK19]. Additionally, for a more vivid comparison of the time efficiency advantages, we also provided regis-

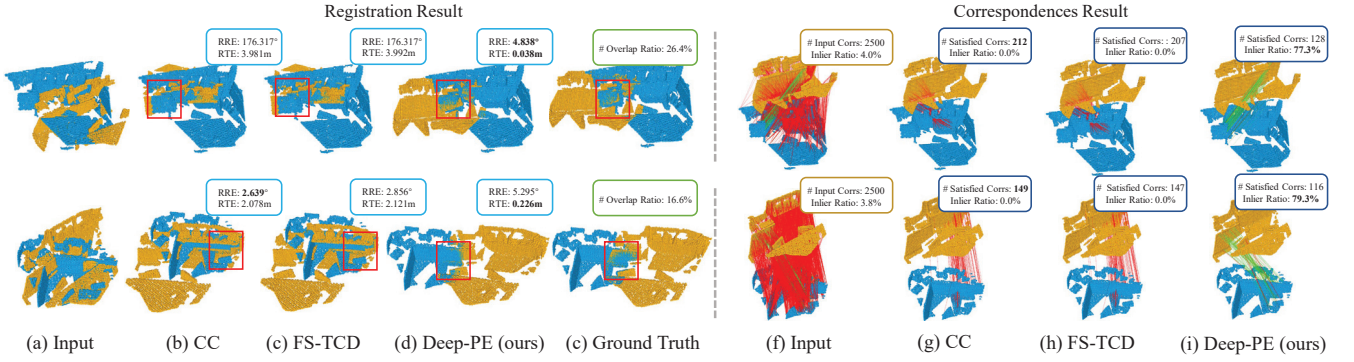


Figure 7: Comparison of registration results between Deep-PE and statistics-based pose evaluators (Under FPFH).

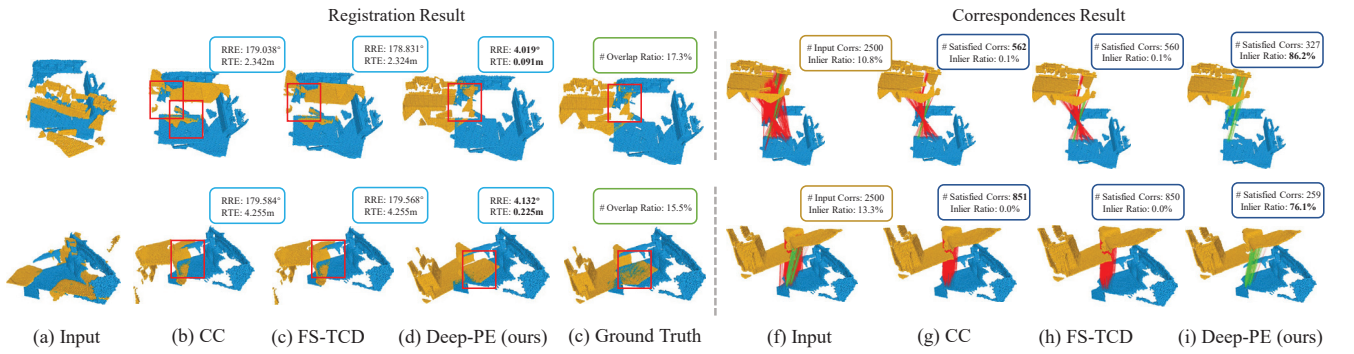


Figure 8: Comparison of registration results between Deep-PE and statistics-based pose evaluators (Under Geotransformer).

tration results and time costs for the classical estimator RANSAC [FB81] and the advanced estimator MAC [ZYZZ23] as references. As shown in Table 2, compared to the baseline evaluator CC [FB81], although FS-TCD [CSY\*23] achieved some performance improvements on 3DMatch, its statistics-based evaluation mechanism still overly relies on the quality of input correspondences, resulting in limited improvements on the challenging 3DLoMatch benchmark with lower inlier ratio. In contrast, our method demonstrated significant enhancements in both 3DMatch and 3DLoMatch benchmarks.

In terms of time overhead, as a pioneering learning-based pose evaluator, we have implemented lightweight and batch processing for candidate poses. Nevertheless, our approach still incurs some time cost compared to traditional pose evaluators. However, considering the significant performance improvements achieved, this is worthwhile. Furthermore, the latest MAC [ZYZZ23] enhances pose estimation accuracy by constructing complex compatibility sub-graphs but at a considerable time expense. In contrast, our method achieves notable performance improvements with less time overhead.

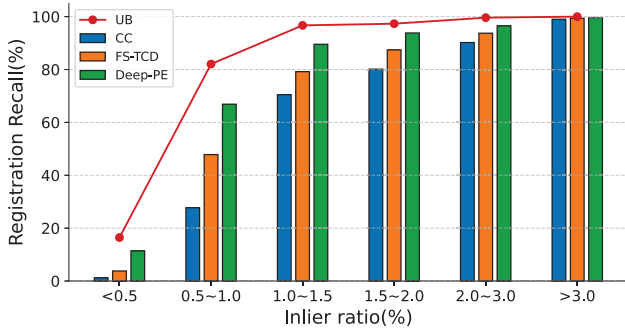
**Robustness to Low Inlier Ratio.** We extracted correspondences using FPFH [RBB09] separately on the 3DMatch and 3DLoMatch datasets, followed by generating candidate poses using SC2-PCR [CSYT22]. Subsequently, based on the inlier ratio, we categorized all point cloud pairs into six different groups. As illustrated in Figure 9 and Figure 3, when the inlier ratio drops below 1%, the RR of

Table 2: Performance boosting for pose estimators when combined with Deep-PE.

	3DMatch RR(%) <sup>↑</sup>		3DLoMatch RR(%) <sup>↑</sup>		Time(s) <sup>↓</sup>
	FPFH	FCGF	FPFH	FCGF	
RANSAC* [FB81]	66.10	91.44	5.50	46.38	2.86
MAC* [ZYZZ23]	84.10	93.72	40.88	59.85	3265.42
SM [LH05] + CC [FB81]	55.88	86.57	12.34	51.43	0.03
PointDSC [BLZ*21] + CC [FB81]	77.57	92.85	20.38	56.09	0.10
SC2-PCR [CSYT22] + CC [FB81]	83.98	93.28	38.57	57.83	0.11
SM [LH05] + FS-TCD [CSY*23]	60.75	87.00	14.74	51.21	0.18
PointDSC [BLZ*21] + FS-TCD [CSY*23]	82.62	93.47	23.09	59.23	0.25
SC2-PCR [CSYT22] + FS-TCD [CSY*23]	87.18	94.15	41.47	61.15	0.26
SM [LH05] + Deep-PE (ours)	63.52	88.46	28.22	63.55	0.64
PointDSC [BLZ*21] + Deep-PE (ours)	85.08	94.21	34.72	67.84	0.71
SC2-PCR [CSYT22] + Deep-PE (ours)	89.40	94.82	50.25	71.87	0.72
	(+7.51)	(+1.36)	(+14.34)	(+11.75)	(+0.61)
	(+4.52)	(+1.54)	(+12.19)	(+14.04)	(+0.61)

statistics-based evaluators [FB81; CSY\*23] is extremely low, while Deep-PE demonstrates commendable performance, highlighting the robustness of our method in scenarios with low inlier ratios.

**Failure Scenes Recognition Capability Analysis.** Compared to traditional statistics-based pose evaluators, our method can identify point cloud pairs within candidate poses lacking correct transformations. We employ SC2-PCR [CSYT22] as the pose es-



**Figure 9:** The registration recall under different inlier ratios on 3DMatch.

timator and then utilize CC [FB81] to pre-select the top 100 most challenging candidate poses for each point cloud pair. In the 3DMatch/3DLoMatch benchmarks, the numbers of failure point cloud pairs generated by PFPF [RBB09], FCGF [CPK19], and Geotransformer (GEO) [QYW\*22] are 187/809, 50/425, and 36/297, respectively. As we treat this issue as a binary classification task, the confidence threshold  $\lambda$  can vary within the range of [0,1] to analyze our method’s ability to identify registration failure scenes. As shown in Table 3, with the increase of the confidence threshold, FSRR also increases correspondingly. This indicates that our predicted confidence scores effectively reflect the likelihood of successful registration. Compared to traditional pose evaluators [FB81; YHQ\*21; CSY\*23], our method avoids the dilemma of selecting from candidate poses lacking correct transformations, which holds significant practical implications.

**Table 3:** Failure scenes recognition recall with different  $\lambda$ .

	3DMatch FSRR(%) $\uparrow$			3DLoMatch FSRR(%) $\uparrow$		
	FPFH	FCGF	GEO	FPFH	FCGF	GEO
$\lambda=0.1$	44.92	4.00	25.00	21.38	9.18	10.44
$\lambda=0.2$	54.55	12.00	27.78	32.51	14.12	12.46
$\lambda=0.3$	60.96	12.00	41.67	40.67	19.53	16.84
$\lambda=0.4$	63.10	20.00	41.67	47.71	26.59	21.21
$\lambda=0.5$	69.52	22.00	44.44	52.41	32.71	26.60
$\lambda=0.6$	73.80	26.00	50.00	55.62	39.06	32.32
$\lambda=0.7$	77.01	32.00	50.00	62.18	44.94	37.37
$\lambda=0.8$	79.68	32.00	50.00	68.23	50.82	45.12
$\lambda=0.9$	84.49	48.00	58.33	76.39	60.47	54.88

**Efficiency with different  $\delta$ .** The advantage of statistics-based pose evaluators over our method lies in computational efficiency, prompting exploration into combining the two approaches. In our model, the ideal scenario involves minimizing the number of candidate poses while ensuring the presence of at least one correct transformation. Therefore, we employ the statistics-based pose evaluator CC [FB81] for candidates’ pre-processing, preventing unnecessary calculations. To provide a deeper understanding of Deep-PE, we present the running time and registration recall results for different values of  $\delta$  in Table 4. As the value of  $\delta$  increases, more candidate poses can be retained, but this comes at the expense of increased computational time. Notably, even when  $\delta > 0.2$ , although this introduces a significant number of incorrect candidate poses, our method maintains stable performance. Setting  $\delta$  to 0.4 (about 400 candidate poses) achieves a balanced trade-off between efficiency and accuracy.

**Table 4:** The running time and registration recall with different  $\delta$ .

	3DMatch RR(%) $\uparrow$			3DLoMatch RR(%) $\uparrow$			Time(s) $\downarrow$
	FPFH	FCGF	GEO	FPFH	FCGF	GEO	
$\delta=0.1$	86.26	94.39	95.81	47.00	68.39	78.81	0.19
$\delta=0.2$	88.42	94.70	95.81	50.00	70.75	79.28	0.34
$\delta=0.3$	88.54	<b>95.07</b>	95.63	50.53	71.25	79.28	0.44
$\delta=0.4$	<b>89.40</b>	<u>94.82</u>	95.93	50.25	71.81	<u>80.23</u>	0.61
$\delta=0.5$	88.97	94.58	<u>96.12</u>	50.42	<u>72.26</u>	80.01	0.67
$\delta=0.6$	89.40	94.52	<u>96.12</u>	50.53	<b>72.71</b>	79.79	0.82
$\delta=0.7$	89.09	94.33	<u>96.06</u>	<u>50.65</u>	71.81	79.79	0.92
$\delta=0.8$	89.34	94.52	95.93	50.31	71.87	79.90	1.07
$\delta=0.9$	88.97	94.58	<u>96.12</u>	50.31	72.09	79.73	1.16
$\delta=1.0$	89.28	94.58	<b>96.24</b>	<b>50.87</b>	72.21	<b>80.24</b>	1.32

#### 4.4. Evaluation on Outdoor Scenes

In Table 5, we compare our method with four statistics-based pose evaluators, including Correspondences Counting (CC) [FB81], Mean Average Error (MAE) [ZYZZ23; YHQ\*21], Mean Square Error (MSE) [YHQ\*21], and Feature and Spatial consistency constrained Truncated Chamfer Distance (FS-TCD) [CSY\*23]. It’s important to note that SC2-PCR [CSYT22] is utilized here to generate candidate poses, and different pose evaluators are employed to select the final transformation. As shown in the table, Deep-PE consistently delivers the best results across all metrics and shares the top spot with PFPF [RBB09] and FCGF [CPK19] descriptor settings, respectively.

**Table 5:** Registration results with different pose evaluators on KITTI.

	FPFH			FCGF		
	RR(%) $\uparrow$	RRE(deg) $\downarrow$	RTE(cm) $\downarrow$	RR(%) $\uparrow$	RRE(deg) $\downarrow$	RTE(cm) $\downarrow$
CC [FB81]	<b>99.64</b>	0.36	7.89	98.20	<b>0.33</b>	20.89
MAE [YHQ*21]	<b>99.64</b>	0.36	7.81	98.20	<b>0.33</b>	20.78
MSE [YHQ*21]	99.34	<b>0.35</b>	<u>7.73</u>	98.20	<b>0.33</b>	<u>20.76</u>
FS-TCD [CSY*23]	<b>99.64</b>	<b>0.35</b>	7.74	98.56	<b>0.33</b>	20.85
Deep-PE (ours)	<b>99.64</b>	<b>0.35</b>	<b>7.68</b>	<b>98.72</b>	<b>0.33</b>	<b>20.74</b>

#### 4.5. Evaluation on Multiway Registration Scenes

To evaluate multi-view registration, we used the enhanced ICL-NUIM dataset [HWMD14]. In the experimental setup, we employed handcrafted descriptors PFPF [RBB09] and advanced Geotransformer [QYW\*22] to extract correspondences, followed by using SC2-PCR [CSYT22] as the pose estimator to generate a set of candidate poses. Here, we compared our method with all known pose evaluators. It is important to note that, to test the model’s generalization ability, we directly used the model trained on 3DMatch without retraining or fine-tuning. As shown in Table 6, due to the performance limitations of the PFPF [RBB09] descriptor, statistics-based pose evaluators are always more significantly impacted. Although FS-TCD [CSY\*23] establishes more correspondences and uses Chamfer distance to evaluate global alignment, it remains overly dependent on the quality of the input correspondences. Since our method fully simulates and learns the alignment state of point clouds and performs confidence prediction through neural network, it is sensitive only to the pose and not reliant on the quality of the input correspondences. Thus, our method achieves the best performance on all scenarios. Under Geotransformer [QYW\*22], our method still maintains certain advantages

across multiple scenarios. It is evident that our method demonstrates strong generalization ability in unknown and more complex scene applications.

**Table 6: Registration results on Augmented ICL-NUIM.**

	FPFH ATE (cm)↓				
	Living1	Living2	Office1	Office2	Mean
CC [FB81]	18.68	14.31	14.63	11.95	14.90
MAE [YHQ*21]	18.36	14.24	14.48	11.82	14.73
MSE [YHQ*21]	18.24	14.12	14.32	11.76	14.61
FS-TCD [CSY*23]	17.56	14.37	13.24	9.49	13.67
Deep-PE (ours)	<b>17.32</b>	<b>14.02</b>	<b>12.94</b>	<b>9.28</b>	<b>13.39</b>
	Geotransformer ATE (cm)↓				
	Living1	Living2	Office1	Office2	Mean
CC [FB81]	17.48	15.26	13.80	9.72	14.07
MAE [YHQ*21]	17.44	15.18	13.78	9.66	14.02
MSE [YHQ*21]	17.40	15.02	13.78	9.52	13.93
FS-TCD [CSY*23]	<b>17.22</b>	15.16	13.42	<b>9.31</b>	13.78
Deep-PE (ours)	<b>17.26</b>	<b>14.62</b>	<b>13.29</b>	<b>9.48</b>	<b>13.67</b>

#### 4.6. Ablation Study of Loss Function

As illustrated in Table 7, we present the experimental results of training Deep-PE using various supervisory signals (e.g., FPFH [RBB09] as the descriptor, SC2-PCR [CSYT22] for generating candidate poses). In this context, the candidate supervisory signals encompass L1(R, t), L1(RMSE), cross-entropy, and the weighted cross-entropy proposed in this paper. Notably, due to substantial variations in pose differences within candidate poses, we observed that directly employing the L1 loss function to train the network to predict rotation and translation errors, or RMSE, presented significant training challenges. The latter, as it predicts a single value, performed slightly better than the former. Treating the problem as a binary classification task yielded superior results, and our proposed weighted cross-entropy, which places emphasis on the pose with the smallest error, achieved optimal performance on both the 3DMatch and 3DLoMatch datasets.

**Table 7: Loss supervision ablation results on the 3DMatch and 3DLoMatch.**

	3DMatch			3DLoMatch		
	RR(%)↑	RRE(deg)↓	RTE(cm)↓	RR(%)↑	RRE(deg)↓	RTE(cm)↓
L1(R,t)	82.78	2.49	6.96	37.29	4.66	12.22
L1(RMSE)	85.52	2.32	6.77	44.14	4.57	11.78
Cross-Entropy	<b>87.12</b>	2.78	7.23	<b>48.66</b>	4.96	12.32
Weighted Cross-Entropy	<b>89.40</b>	<b>2.18</b>	<b>6.52</b>	<b>50.25</b>	<b>4.32</b>	<b>11.24</b>

#### 4.7. Ablation Study of Modules

In this section, we carry out comprehensive ablation studies to gain a deeper understanding of the different modules within our approach. We employ Geotransformer [QYW\*22] to generate correspondences on both 3DMatch and 3DLoMatch datasets, while SC2-PCR [CSYT22] with CC [FB81] serves as the baseline, as depicted in Row (1) of Table 8.

**Feature Extractor with Pre-Train.** We utilize the pre-trained Geotransformer [QYW\*22] model as our feature extraction module and refrain from participating in subsequent network updates. This choice is motivated by our aim to maintain a stable point-wise

**Table 8: Ablation experiments of main modules on 3DMatch and 3DLoMatch.**

	PT	GEO	PAA	PCP	3DMatch			3DLoMatch		
					RR(%)↑	RRE(deg)↓	RTE(cm)↓	RR(%)↑	RRE(deg)↓	RTE(cm)↓
(1)	×	×	×	×	83.98	2.24	6.80	38.57	<b>4.16</b>	<b>10.23</b>
(2)	×	✓	✓	✓	-	-	-	-	-	-
(3)	✓	×	✓	✓	87.92	<b>2.12</b>	6.69	47.45	4.27	10.92
(4)	✓	✓	×	✓	81.21	2.34	6.92	33.18	4.89	10.78
(5)	✓	✓	×	×	86.63	2.22	6.75	44.47	4.22	10.37
(6)	✓	✓	✓	✓	<b>89.40</b>	<b>2.18</b>	<b>6.52</b>	<b>50.25</b>	4.32	11.24

feature representation of the input point clouds. Nevertheless, we also attempted to train Deep-PE in an end-to-end manner; however, the network did not converge, as shown in Row (2). Our analysis indicates that the loss function used solely for supervising pose confidence does not effectively guide the generation of point-wise features. Furthermore, due to limitations in our network structure, certain loss functions for supervising feature matching, such as circle loss [QYW\*22], could not be effectively adapted. As mentioned earlier, for the feature extraction module, we opt for the pre-training mode over the end-to-end approach.

**Performance of Feature Extractor.** To investigate the influence of feature description quality on our results, we conducted an experiment where we replaced the geometric self-attention module in Geotransformer [QYW\*22] with the vanilla self-attention module [VSP\*17], keeping other modules unchanged. We then re-trained the model with this modification. As depicted in Row (3) and Row (6), the RR obtained using the geometric self-attention module is 1.48% higher than that achieved with the vanilla self-attention module on the 3DMatch dataset and 2.80% higher on the 3DLoMatch dataset. This indicates that improved feature description can result in enhanced performance of our method.

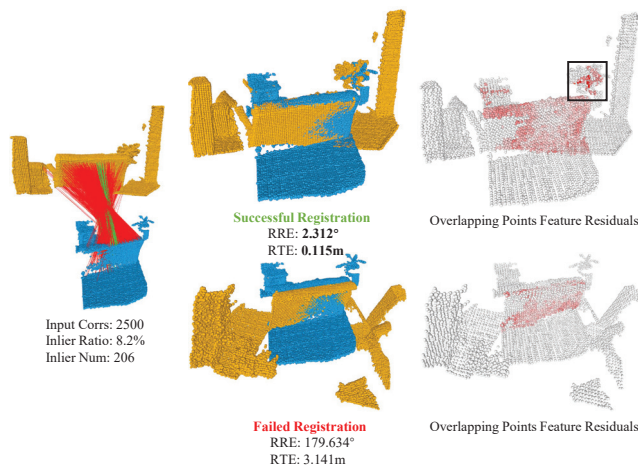
**Pose-Aware Attention Module.** Being the most pivotal module in our approach, feature updating is accomplished through local attention operations under specific poses, resulting in distinct feature representations for different poses. However, during module ablation experiments, all feature representations become independent of pose, introducing ambiguity and hindering network convergence. To underscore the significance of this module, we conducted an experiment where we directly sought nearest neighbors under different poses to generate feature residuals, serving as a replacement for the attention module. The results, as shown in the complete model (Row 6) compared to Row (4), reveal that the RR obtained using the pose-aware attention module is 8.19% higher than when using the direct feature residual approach on the 3DMatch dataset and 17.07% higher on the 3DLoMatch dataset. This underscores that the module effectively learns the degree of feature matching under different poses through an attention mechanism, rather than relying solely on simple residuals.

**Pose Confidence Prediction Module.** We utilize a single linear layer and a sigmoid layer to directly process the learned feature residuals, deviating from the design in this paper. Comparing Row (5) and Row (6), it becomes evident that employing a multi-layer perceptron for additional feature learning enhances the RR by 2.77% on the 3DMatch dataset and 5.78% on the 3DLoMatch dataset. This underscores the importance of further refining the acquired feature residuals through additional learning.

## 5. Limitations

Deep-PE relies on the candidate poses generated by the pose estimator and then finds the correct transformation from them, replacing the existing statistics-based pose evaluation mechanism. However, when no correct transformations exist in the candidate poses, our method cannot generate a correct transformation, which is the upper bound of our method.

In addition, we present a failure case, as illustrated in Figure 10. Under the correct pose, the point cloud in the upper right corner exhibits severe omissions, resulting in blurred features and the failure to establish effective correspondences. However, this region represents the overlapping area under the correct pose, where the features of the planar region points exhibit minimal differences, leading to the establishment of numerous correspondences. Due to our model considering only the pose with overlapping planar regions as the optimal transformation, this scenario is unavoidable.



**Figure 10:** A failure example. In this scenario, a significant number of correspondences are identified within planar regions, owing to the similarity of features. However, the transformation derived from these correspondences is incorrect.

## 6. Conclusion

In this paper, we introduce Deep-PE as a pioneering approach that leverages deep learning to determine the optimal pose without necessitating explicit initial correspondences. The architecture of Deep-PE is meticulously crafted and lightweight, featuring core components including a pose-aware attention module that simulates and learns alignment status under various candidate poses, and a pose confidence prediction module that predicts the confidence for each candidate pose, ultimately selecting the pose with the highest confidence score as the final transformation. Moreover, we propose a weighted cross-entropy loss to reweight the loss of candidate poses based on their similarity to the ground truth, thereby enhancing the capability to select the optimal pose. Our experiments across multiple public benchmarks demonstrate that Deep-PE achieves state-of-the-art results and showcases commendable performance in low-overlap scenarios with poor input correspondence quality, such as those encountered in 3DLoMatch.

## References

- [AHY\*20] AO, SHENG, HU, QINGYONG, YANG, BO, et al. “SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration”. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 11748–11757 2, 3.
- [BLZ\*20] BAI, XUYANG, LUO, ZIXIN, ZHOU, LEI, et al. “D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features”. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 6358–6366 2.
- [BLZ\*21] BAI, XUYANG, LUO, ZIXIN, ZHOU, LEI, et al. “PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency”. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 15854–15864 2, 3, 6, 7, 14.
- [BM17] BARÁTH, DÁNIEL and MATAS, JIRI. “Graph-Cut RANSAC”. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), 6733–6741 2, 3.
- [BM92] BESL, PAUL J. and MCKAY, NEIL D. “A Method for Registration of 3-D Shapes”. *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992), 239–256. URL: <https://api.semanticscholar.org/CorpusID:218743463>.
- [CDK20] CHOY, CHRISTOPHER BONGSOO, DONG, WEI, and KOLTUN, VLADLEN. “Deep Global Registration”. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 2511–2520 2, 3, 6.
- [CPK19] CHOY, CHRISTOPHER BONGSOO, PARK, JAESIK, and KOLTUN, VLADLEN. “Fully Convolutional Geometric Features”. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 8957–8965 2, 3, 6, 8, 14.
- [CSY\*23] CHEN, ZHI, SUN, KUN, YANG, FAN, et al. “SC2-PCR++: Rethinking the Generation and Selection for Efficient and Robust Point Cloud Registration”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023) 2, 3, 5–9.
- [CSYT22] CHEN, ZHI, SUN, KUN, YANG, FAN, and TAO, WENBING. “Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 13221–13231 2, 3, 5–9, 12–14.
- [FB81] FISCHLER, MARTIN A. and BOLLES, ROBERT C. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. *Commun. ACM* 24 (1981), 381–395 2, 3, 6–9, 12, 13.
- [GLU12] GEIGER, ANDREAS, LENZ, PHILIP, and URTASUN, RAQUEL. “Are we ready for autonomous driving? The KITTI vision benchmark suite”. *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), 3354–3361 14.
- [GZWW18] GOJCIC, ZAN, ZHOU, CAIFA, WEGNER, JAN DIRK, and WIESER, ANDREAS. “The Perfect Match: 3D Point Cloud Matching With Smoothed Densities”. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 5540–5549 2.
- [HGU\*21] HUANG, SHENGYU, GOJCIC, ZAN, USVYATSOV, MIKHAIL, et al. “Predator: Registration of 3d point clouds with low overlap”. *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2021, 4267–4276 2, 3, 5, 13, 14.
- [HMZ20] HUANG, XIAOSHUI, MEI, GUOFENG, and ZHANG, JIAN. “Feature-Metric Registration: A Fast Semi-Supervised Approach for Robust Point Cloud Registration Without Correspondences”. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 11363–11371 3.
- [HWMD14] HANDA, ANKUR, WHELAN, THOMAS, McDONALD, JOHN B., and DAVISON, ANDREW J. “A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM”. *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), 1524–1531. URL: <https://api.semanticscholar.org/CorpusID:206850587> 8, 14.

- [HYFY20] HE, YIHUI, YAN, RUI, FRAGKIADAKI, KATERINA, and YU, SHOOU-I. “Eipolar transformers”. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, 7779–7788 3.
- [JDW\*23] JIANG, HAOBO, DANG, ZHENG, WEI, ZHEN, et al. “Robust outlier rejection for 3d registration with variational bayes”. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, 1148–1157 2.
- [JV10] JIAN, BING and VEMURI, BABA C. “Robust point set registration using gaussian mixture models”. *IEEE transactions on pattern analysis and machine intelligence* 33.8 (2010), 1633–1645 3.
- [LH05] LEO, MARIUS and HEBERT, MARTIAL. “A spectral technique for correspondence problems using pairwise constraints”. *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1 2* (2005), 1482–1489 Vol. 2, 2, 3, 6, 7.
- [LPL21] LI, XUEQIAN, PONTES, JHONY KAESEMODEL, and LUCEY, SIMON. “PointNetLK Revisited”. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 12758–12767 3, 4.
- [LWZ\*19] LU, WEIXIN, WAN, GUOWEI, ZHOU, YAO, et al. “Deepvcp: An end-to-end deep neural network for point cloud registration”. *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, 12–21 2.
- [MTH\*23] MEI, GUOFENG, TANG, HAO, HUANG, XIAOSHUI, et al. “Unsupervised deep probabilistic approach for partial point cloud registration”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 13611–13620 3.
- [MTN\*02] MYATT, DARREN R., TORR, PHILIP H. S., NASUTO, SLAWOMIR JAROSLAW, et al. “NAPSAC: High Noise, High Dimensional Robust Estimation - it’s in the Bag”. *British Machine Vision Conference*. 2002 3.
- [PL00] PAPADOPOULOU, THÉODORE and LOURAKIS, MANOLIS I. A. “Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications”. *European Conference on Computer Vision*. 2000. URL: <https://api.semanticscholar.org/CorpusID:1327133>.
- [PMR\*19] PAIS, G. DIAS, MIRALDO, PEDRO, RAMALINGAM, SRIKUMAR, et al. “3DRegNet: A Deep Neural Network for 3D Point Registration”. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 7191–7201 2, 3.
- [QSMG16] QI, C., SU, HAO, MO, KAICHUN, and GUIBAS, LEONIDAS J. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 77–85. URL: <https://api.semanticscholar.org/CorpusID:5115938> 3, 4.
- [QYW\*22] QIN, ZHENG, YU, HAO, WANG, CHANGJIAN, et al. “Geometric Transformer for Fast and Robust Point Cloud Registration”. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 11133–11142 2–4, 6, 8, 9, 12–14.
- [RBB09] RUSU, RADU BOGDAN, BLODOW, NICO, and BEETZ, MICHAEL. “Fast Point Feature Histograms (FPFH) for 3D registration”. *2009 IEEE International Conference on Robotics and Automation* (2009), 3212–3217 2, 6–9.
- [SSW\*21] SUN, JIAMING, SHEN, ZEHONG, WANG, YUANG, et al. “LoFTR: Detector-Free Local Feature Matching with Transformers”. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 8918–8927 3.
- [TQD\*19] THOMAS, HUGUES, QI, C., DESCHAUD, JEAN-EMMANUEL, et al. “KPConv: Flexible and Deformable Convolution for Point Clouds”. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 6410–6419 3, 4.
- [TSdS10] TOMBARI, FEDERICO, SALTI, SAMUELE, and di STEFANO, LUIGI. “Unique Signatures of Histograms for Local Surface Description”. *European Conference on Computer Vision*. 2010 2.
- [VSP\*17] VASWANI, ASHISH, SHAZEER, NOAM M., PARMAR, NIKI, et al. “Attention is All you Need”. *ArXiv abs/1706.03762* (2017) 3, 9.
- [WLH\*23] WANG, HAIPING, LIU, YUAN, HU, QINGYONG, et al. “RoReg: Pairwise Point Cloud Registration with Oriented Descriptors and Local Rotations.” *IEEE transactions on pattern analysis and machine intelligence* PP (2023) 2, 3.
- [WPN\*19] WANG, ALEX, PRUKSACHATKUN, YADA, NANGIA, NIKITA, et al. “SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems”. *ArXiv abs/1905.00537* (2019). URL: <https://api.semanticscholar.org/CorpusID:143424870>.
- [WS19] WANG, YUE and SOLOMON, JUSTIN M. “Deep Closest Point: Learning Representations for Point Cloud Registration”. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 3522–3531 3.
- [WSL\*19] WANG, YUE, SUN, YONGBIN, LIU, ZIWEI, et al. “Dynamic graph cnn for learning on point clouds”. *ACM Transactions on Graphics (tog)* 38.5 (2019), 1–12 3.
- [YEK\*20] YUAN, WENTAO, ECKART, BENJAMIN, KIM, KIHWAN, et al. “DeepGMR: Learning Latent Gaussian Mixture Models for Registration”. *European Conference on Computer Vision*. 2020. URL: <https://api.semanticscholar.org/CorpusID:221186725>.
- [YHQ\*21] YANG, JIAQI, HUANG, ZHIQIANG, QUAN, SIWEN, et al. “Toward Efficient and Robust Metrics for RANSAC Hypotheses and 3D Rigid Registration”. *IEEE Transactions on Circuits and Systems for Video Technology* 32 (2021), 893–906. URL: <https://api.semanticscholar.org/CorpusID:233804557> 2, 3, 6, 8, 9.
- [YL20] YEW, ZI JIAN and LEE, GIM HEE. “RPM-Net: Robust Point Matching Using Learned Features”. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 11821–11830 3.
- [YLS\*21] YU, HAO, LI, FU, SALEH, MAHDI, et al. “CoFiNet: Reliable Coarse-to-fine Correspondences for Robust Point Cloud Registration”. *Neural Information Processing Systems*. 2021 2, 3.
- [ZSN\*16] ZENG, ANDY, SONG, SHURAN, NIESSNER, MATTHIAS, et al. “3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions”. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 199–208 12, 14.
- [ZYZZ23] ZHANG, XIYU, YANG, JIAQI, ZHANG, SHIKUN, and ZHANG, YANNING. “3D Registration with Maximal Cliques”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 17745–17754 2, 3, 6–8.

## Appendix

### A. More Insights into Pose Evaluation

In this section, we provide insights into our focus on the pose evaluation stage and outline the essential criteria for an effective pose evaluator.

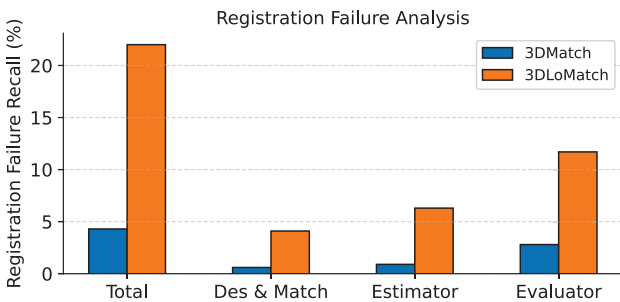
#### A.1. Pose Evaluator as the Bottleneck

A typical estimator-based registration pipeline consists of three fundamental steps: feature description and matching, pose estimation, and pose evaluation. In the subsequent sections, we aim to identify the bottleneck step.

*Experimental setting.* We selected Geotransformer [QYW\*22] for correspondence extraction and SC2-PCR [CSYT22] as the pose estimator to generate a set of candidate poses. Our experiments were performed on the 3DMatch and 3DLoMatch datasets [ZSN\*16]. Using Geotransformer [QYW\*22] and SC2-PCR [CSYT22], we generated 5000 correspondences and produced 1000 candidate poses. We then applied the widely used Correspondence Counting (CC) as evaluator [FB81] to select the final pose.

*Indicators.* We utilize the following indicators to analyze registration failures:

- **Total:** This indicator represents the overall recall of registration failures.
- **Des & Match:** This indicator represents the percentage of cases where deteriorated descriptor performance results in too few correct correspondences (less than three), emphasizing that at least three correct correspondences are essential for accurate pose estimation.
- **Estimator:** This indicator quantifies the proportion of instances where, despite having a minimum of three correct correspondences, the candidate poses generated by the pose estimator fail to include a least one correct transformation.
- **Evaluator:** This indicator shows the percentage of cases where the correct transformations, although present among the candidate poses, is incorrectly overlooked by the pose evaluator in favor of an incorrect final pose selection.

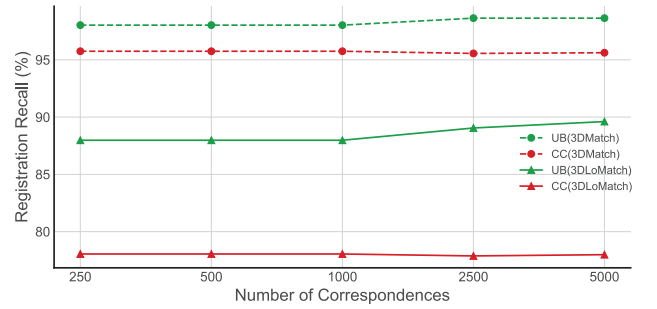


**Figure A:** Utilizing 3DMatch and 3DLoMatch datasets, we plot the registration failure recall across three stages: Des & Match (correspondences generation), Estimator (candidate poses estimation), and Evaluator (optimal pose selection).

*Conclusion.* Analyzing the statistics of registration failure recall

for each step, as illustrated in Figure A, reveals a significant finding: up to 12% of failures can be attributed to the pose evaluator, while failures due to feature descriptors (Des & Match) are limited to approximately 4%. This observation highlights that the primary bottleneck in the current estimator-based registration pipeline lies with the statistics-based pose evaluators, which impede correct pose selection.

#### A.2. Performance Degradation of Statistics-based Evaluators in Low Overlap



**Figure B:** The performance of Correspondence Counting (CC) significantly declines on 3DLoMatch, falling well below the upper bound (UB) values. Here, UB is defined by the likelihood that at least one correct transformation appears within the candidate poses.

In this part, we come to examine the scalability of registration performance for statistics-based pose evaluators in relation to varying numbers of correspondences. The experimental setup remained consistent with the previous subsection, with the distinction of employing the Geotransformer [QYW\*22] to generate different quantities of correspondences. Results from experiments on the 3DMatch and 3DLoMatch datasets are depicted in Figure B. On one side, the likelihood of Correspondence Counting (CC) [FB81] failing increases in scenarios of low overlap. As shown in Table A, this is particularly evident when the overlap region is either texture-deficient or has a low overlap rate, leading to correspondences that are significantly marred by erroneous outliers. On the opposite spectrum, an increase in the number of correspondences does not necessarily translate to an improved registration recall. In fact, a reduction in the registration recall rate was observed when the sampling numbers reached 2500 and 5000, subsequently impairing the efficacy of CC [FB81] in identifying the correct transformation. We also observe that with an increase in the number of correspondences samples, the upper bound (UB) values correspondingly increase. This indicates that the pose estimator can increase the probability of including the correct transformations in the candidate poses. However, the registration recall decreases due to the statistical-based pose evaluator, highlighting the necessity of proposing an evaluator independent of the quality of input correspondences.

#### A.3. Essential Criteria for an Effective Pose Evaluator

Based on the above discussion, we give essential criteria for an effective pose evaluator.

**Table A:** Inlier ratio across different correspondence sample numbers.

# Samples	3DMatch IR (%) <sup>↑</sup>					3DLoMatch IR (%) <sup>↑</sup>				
	250	500	1000	2500	5000	250	500	1000	2500	5000
Geotransformer [QYW*22]	85.1	82.2	76.0	75.2	71.9	57.7	52.9	46.2	45.3	43.5

**Lightweight Network Design.** Typically, traditional pose estimation algorithms like RANSAC [FB81] tend to generate a large pool of candidate poses. While some robust estimators, such as SC2-PCR [CSYT22], are designed to yield a smaller, more precise set of candidate poses, the overall number of these poses remains considerable. Consequently, the development of a versatile, learning-based pose evaluator becomes critical for the efficient evaluation of each candidate pose. This calls for a lightweight network architecture that not only processes candidate poses efficiently but also supports batch processing for enhanced throughput.

**Perception of Global Alignment.** The network is tasked with learning to accurately differentiate between correct and incorrect poses by incorporating global alignment information. This differentiation is achievable through the use of statistics-based pose estimators, which analyze the overall spatial relationships and alignment patterns across the entire scene. By focusing on these global characteristics, the network can more effectively identify the validity of each pose, distinguishing those that align well with the global context from those that do not.

**Pose Sensitivity.** The network needs to demonstrate sensitivity to variations in poses, requiring it not only to distinguish accurately between correct and incorrect pose hypotheses but also to assess the degrees of accuracy among candidate poses. Specifically, the network should assign a higher confidence score to candidate poses that are exact in their accuracy compared to correct ones that present a degree of deviation. Implementing this capability is crucial for improving the overall effectiveness of pose evaluation, ensuring that the most accurate poses are prioritized and selected.

**Quality Independent of Input Correspondences.** The design of the network should prioritize sensitivity to pose variations above all, especially given that a low inlier count among input correspondences can lead to situations where, despite outliers being incorrect, they still demonstrate compatibility and advocate for the same transformation. In these scenarios, statistics-based strategies might not be effective. As depicted in Figure A, even robust pose estimators are capable of generating at least one correct pose under such conditions. Consequently, our methodology primarily focuses on evaluating the candidate poses themselves, rather than the quality of the input correspondences. This approach ensures that our proposed network maintains strong robustness across a wide range of conditions, effectively handling the intricacies and challenges posed by various scenarios.

**Capability to Identify Incorrect poses** Traditional pose evaluators primarily rely on statistical mechanisms to select the optimal transformation from a set of candidate poses, but they have limitations in recognizing incorrect poses. For instance, in scenarios where the candidate poses lacks the presence of correct transformations, these evaluators may have to choose an incorrect pose as the

final transformation. The significance of a versatile pose evaluator lies in its ability to identify incorrect poses, which holds practical value in real-world scenarios. This not only enhances the reliability of the pose estimation process but also substantially reduces the risk of propagating errors, especially in applications where accurate pose estimation is crucial.

## B. Evaluation Metrics

Following common practice [HGU\*21; QYW\*22], we use different evaluation metrics to evaluate the quality of correspondences and registration results.

**Inlier Ratio (IR).** It measures the ratio of putative correspondences whose residual distance is smaller than a threshold (i.e.  $\tau_1 = 0.1/0.6m$  for indoor/outdoor scenes) under the ground-truth transformation  $\hat{\mathbf{T}}_{\mathbf{P} \rightarrow \mathbf{Q}}$ :

$$\mathbf{IR} = \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{p}_{x_i}, \mathbf{q}_{y_i}) \in \mathcal{C}} \llbracket \|\hat{\mathbf{T}}_{\mathbf{P} \rightarrow \mathbf{Q}}(\mathbf{p}_{x_i}) - \mathbf{q}_{y_i}\|_2 < \tau_1 \rrbracket \quad (5)$$

where  $\llbracket \cdot \rrbracket$  is the Iverson bracket.

**Relative Rotation Error (RRE).** It measures the geodesic distance in degrees between estimated and ground-truth rotation matrices.

$$\mathbf{RRE} = \arccos \left( \frac{\text{trace}(\mathbf{R}^T \cdot \hat{\mathbf{R}} - 1)}{2} \right) \quad (6)$$

**Relative Translation Error (RTE).** It measures the Euclidean distance between estimated and ground-truth translation vectors.

$$\mathbf{RTE} = \|\mathbf{t} - \hat{\mathbf{t}}\|_2 \quad (7)$$

**Registration Recall (RR).** It is defined as the fraction of the point cloud pairs whose RRE and RTE are both below certain thresholds:

$$\mathbf{RR} = \frac{1}{M} \sum_{i=1}^M \llbracket \mathbf{RRE}_i < \tau_2 \wedge \mathbf{RTE}_i < \tau_3 \rrbracket \quad (8)$$

$(\tau_2, \tau_3)$  is set to  $(15^\circ, 0.3m)$  and  $(5^\circ, 0.6m)$  for the indoor and outdoor scenes, respectively.

**Failure Scenes Recognition Recall (FSRR).** We propose a novel metric to assess the capability of a pose evaluator in identifying the proportion of point cloud pairs in candidate poses that do not contain the correct transformation.

$$\mathbf{FSRR} = \frac{1}{N} \sum_{i=1}^N \llbracket \sum_{j=1}^K \llbracket S_j < \lambda \rrbracket = K \rrbracket \quad (9)$$

where  $N$  represents the number of point cloud pairs in candidate pose hypotheses that do not contain the correct transformation,  $K$  denotes the number of candidate poses for each point cloud pair,  $S$  represents the confidence scores predicted by the pose evaluator, and  $\lambda$  is the truncation threshold. The metric indicates the ratio of point cloud pairs, among the total, for which the confidence predictions of all  $K$  candidate poses are below the threshold  $\lambda$ .

## C. More Implementation Details

### C.1. Datasets

**Indoor scenes.** We use the 3DMatch benchmark [ZSN\*16] for evaluating the performance on indoor scenes. It contains 1623 pairs of point clouds with ground-truth camera poses, which are obtained by 8 different RGBD sequences. For each pair of point clouds, we downsample them using a voxel size of 5cm. Then, we use different descriptors and pose estimators to extract correspondences and candidate poses, respectively. Partial overlapping is challenging in point cloud registration. In order to further test the performance of our method, 3DLoMatch benchmark [HGU\*21] is adopted to further verify the performance of the algorithm on low-overlapped point cloud registration. It contains 1781 pairs of point clouds with overlap ratios ranging from 10% to 30%.

**Outdoor scenes.** We use the KITTI benchmark [GLU12] for evaluating the performance on outdoor scenes. It contains 11 outdoor driving scenarios of point clouds. Following [CPK19], we choose the 8 to 10 scenarios as test datasets. For all the LIDAR scans, we use the first scan that is taken at least 10cm apart within each sequence to create a pair, which can obtain 555 pairs of point clouds for testing.

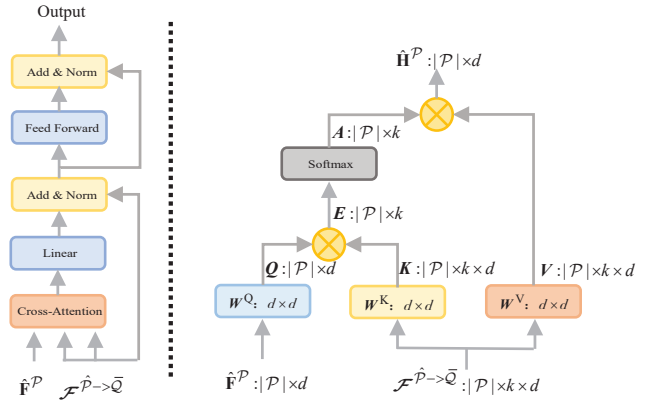
**Multway registration scenes.** We use the ICL-NUIM augments benchmark [HWMD14] for evaluating the performance on multiway scenes. It consists of four camera trajectories from two scenes for testing. Following [BLZ\*21; CSYT22], we fuse 50 consecutive RGBD frames to generate the point cloud fragments.

### C.2. Network Architecture Details

**Feature Extractor.** We utilized the pre-trained GeoTransformer as the backbone network for feature extraction. Notably, to achieve a lightweight design, we only utilize two layers of points and their corresponding features. Specifically, for the 3DMatch and KITTI datasets, we selected the coarsest layer and the penultimate layer, with feature dimensions of 256 and 512, respectively. For detailed network architecture, please refer to the original implementation. To further learn the features and facilitate subsequent computations, we employed two linear layers, Linear1 (256->256) and Linear2 (512->256), to map the obtained features to the same dimension. Specifically, the points and their updated features at the coarsest layers are represented as  $\hat{\mathcal{P}}/\hat{\mathcal{Q}}$  and  $\hat{\mathbf{F}}^{\mathcal{P}}/\hat{\mathbf{F}}^{\mathcal{Q}} \in \mathbb{R}^{|\hat{\mathcal{P}}/\hat{\mathcal{Q}}| \times d}$ , while those at the penultimate layers are represented as  $\tilde{\mathcal{P}}/\tilde{\mathcal{Q}}$  and  $\tilde{\mathbf{F}}^{\mathcal{P}}/\tilde{\mathbf{F}}^{\mathcal{Q}} \in \mathbb{R}^{|\tilde{\mathcal{P}}/\tilde{\mathcal{Q}}| \times d}$  ( $d = 256$ ).

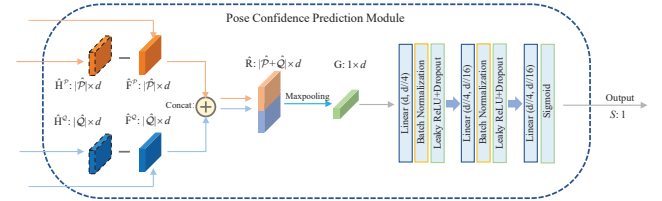
**Pose-Aware Attention Module.** We adopted a multi-head attention mechanism with the number of attention heads set to 4. We flexibly adjusted the attention regions based on different candidate poses to obtain the corresponding feature volume  $\mathcal{F}^{\hat{\mathcal{P}} \rightarrow \tilde{\mathcal{Q}}} \in \mathbb{R}^{|\hat{\mathcal{P}}| \times k \times d}$  for  $\hat{\mathbf{F}}^{\mathcal{P}}$ . The computation of the pose-aware attention features  $\hat{\mathbf{H}}^{\mathcal{P}}$  for  $\hat{\mathbf{F}}^{\mathcal{P}}$  is shown in Figure C, and the  $\hat{\mathbf{H}}^{\mathcal{Q}}$  is computed in the same way.

**Pose Confidence Prediction Module.** As shown in Figure D, to capture the differentiation in feature updates under different poses, we apply a residual operation to the obtained features in this context:  $\hat{\mathbf{R}}^{\hat{\mathcal{P}}/\tilde{\mathcal{Q}}} = \hat{\mathbf{H}}^{\hat{\mathcal{P}}/\tilde{\mathcal{Q}}} - \hat{\mathbf{F}}^{\hat{\mathcal{P}}/\tilde{\mathcal{Q}}}$ . Then we concatenate the obtained residual features:  $\text{Cat}(\hat{\mathbf{R}}^{\hat{\mathcal{P}}}, \hat{\mathbf{R}}^{\tilde{\mathcal{Q}}}) \in \mathbb{R}^{|\hat{\mathcal{P}}+\tilde{\mathcal{Q}}| \times d}$  and then perform



**Figure C:** Left: The structure of pose-aware attention module. Right: The computation graph of pose-aware attention.

a max-pooling operation to obtain the global feature:  $\mathbf{G} \in \mathbb{R}^{1 \times d}$ . Subsequently, we employ a 3-layer MLP with 256->64->16->1 channels to map and obtain the corresponding confidence scores  $S$ . Following the first two linear layers, each layer is succeeded by 1D batch normalization, a leaky ReLU, and a dropout layer. Since we consider the problem as binary classification, we apply the sigmoid layer after the final linear layer.



**Figure D:** The structure of Pose Confidence Prediction Module.

### C.3. Training Details

We train indoor and outdoor Deep-PE models on 3DMatch and KITTI datasets. To generate training and validation sets, we first use pre-trained indoor and outdoor Geotransformer [QYW\*22] models to extract correspondences. We then use SC2-PCR [CSYT22] as the pose estimator to extract 1000 candidate poses for each point cloud pair. In this scenario, we observed instances where correct or incorrect poses were absent. To ensure an adequate presence of both correct and incorrect poses in the candidate pose pool, we introduced perturbations by randomly sampling small rotations ( $Rot \in [0^\circ, 15^\circ]$  for 3DMatch,  $Rot \in [0^\circ, 5^\circ]$  for KITTI) or large rotations ( $Rot \in [15^\circ, 60^\circ]$  for 3DMatch,  $Rot \in [5^\circ, 20^\circ]$  for KITTI) and multiplying them with the ground truth relative pose. This process aims to augment the pool with a diverse set of correct and incorrect poses. For each candidate pose, we calculate the corresponding Root Mean Square Error (RMSE) according to the ground-truth correspondences.

**Table B:** Time and CUDA memory usage of inference under different batch sizes.

	batchsize			
	1	20	40	60
Average time (s)	2.26	1.82	1.54	1.32
Cuda memory (MB)	1420	6790	12810	18950

#### C.4. Time and memory expense

As shown in Table B, we present the inference time and CUDA memory usage under different batch sizes. The results indicate that, due to our lightweight design, batch inference is feasible within the available memory, significantly accelerating the model's inference speed.