Distinctiveness oriented Positional Equilibrium for Point Cloud Registration

Taewon Min* KAIST
Chonghyuk Song* CMU
Eunseok Kim*
Inwook Shim† ADD

Abstract

*Recent state-of-the-art learning-based approaches to point cloud registration have largely been based on graph neural networks (GNN). However, these prominent GNN backbones suffer from the indistinguishable features problem associated with oversmoothing and structural ambiguity of the high-level features, a crucial bottleneck to point cloud registration that has evaded scrutiny in the recent relevant literature. To address this issue, we propose the **D**istinctiveness **o**riented **P**ositional **E**quilibrium (**DoPE**) module, a novel positional embedding scheme that significantly improves the distinctiveness of the high-level features within both the source and target point clouds, resulting in superior point matching and hence registration accuracy. Specifically, we use the DoPE module in an iterative registration framework, whereby the two point clouds are gradually registered via rigid transformations that are computed from DoPE's position-aware features. With every successive iteration, the DoPE module feeds increasingly consistent positional information to would-be corresponding pairs, which in turn enhances the resulting point-to-point correspondence predictions used to estimate the rigid transformation. Within only a few iterations, the network converges to a desired equilibrium, where the positional embeddings given to matching pairs become essentially identical. We validate the effectiveness of DoPE through comprehensive experiments on various registration benchmarks, registration task settings, and prominent backbones, yielding unprecedented performance improvement across all combinations.*

1. Introduction

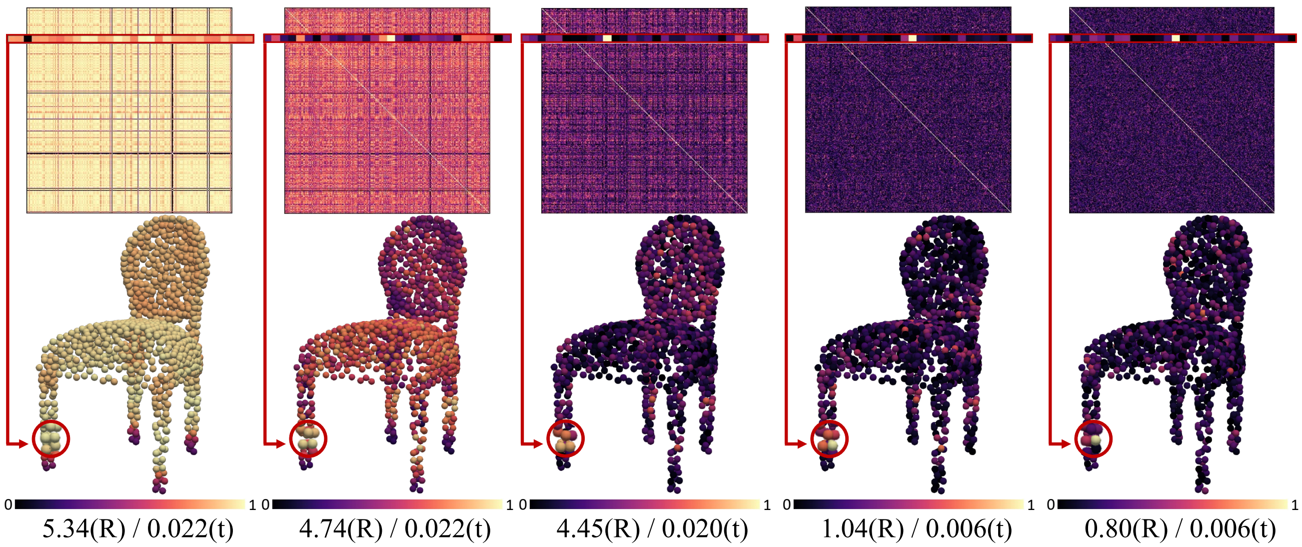
Point cloud registration is a well-known task by which two point clouds are matched via a rigid transformation. For a source point cloud $\mathcal{X}$ and a target point cloud $\mathcal{Y}$, the registration problem is finding a rigid transformation that minimizes the geometric shape differences between $\mathcal{Y}$ and the transformed $\mathcal{X}$. In many applications such as 3D reconstruction and simultaneous localization and mapping (SLAM), the registration process has long relied on traditional, non-learning-based algorithms to predict the optimal rigid transformations.

Recently, deep learning methods have brought remarkable advances in a variety of 3D vision tasks, ranging from classification, segmentation, and, point cloud registration. A common theme among many learning-based registration methods [18, 19, 24, 5, 25] is the fact they are comprised of 1) a feature extraction backbone, usually a graph neural network (GNN), which generates per-point feature descriptors via iterative local aggregation, followed by 2) a feature matching step, which computes point-to-point matchability scores, or (soft) correspondences, between the source and target point clouds using their extracted features.

For example, Deep Closest Point (DCP) [18] computes point correspondences from *learned* features, via attention combined with pointer generation, in order to desensitize the network from initialization and avoid local minima. RPM-Net [24] incorporates Robust Point Matching (RPM) [3] into the feature matching step to be able to also handle outliers and missing correspondences. On the other hand, DeepGMR [25] avoids exhaustive point-to-point correspondences all together by learning correspondences from both point clouds to a common distribution inside a learned latent space.

While these recent methods have made significant improvements to the feature matching step and displayed state-of-the-art performance, they overlook a key design consideration for feature extraction that can critically affect registration accuracy: the distinctiveness of the per-point features within both the source and target point clouds; that is, in order to obtain accurate point-to-point correspondences for estimating the optimal rigid transformation, the desired point features should sufficiently represent the geometric pattern in the neighborhood of any given point *while still being distinguishable enough from the local patterns surrounding other points within the same point cloud.* However, many of the GNN backbones typically used to embed the input point clouds into the feature space [12, 20] are susceptible to oversmoothing [6, 17, 1] and structural

*This work was done while Taewon Min, Chonghyuk Song, and Eunseok Kim were with the Agency for Defense Development (ADD).
†Corresponding author: iwshim@add.re.kr



(a) GNNs only (b) Local attention (c) Non-local attention (d) *DoPE** (e) *DoPE***

Figure 1: Visualization of indistinguishable features problem with cosine similarity scores in the feature space. The upper-row figures represent the self-similarity matrices of the source point cloud; those in the bottom row visualize the similarity scores, in 3D space, between the point in the red enclosed circle with all points that comprise the chair. *DoPE** and *DoPE*** use local and non-local attention, respectively, for feature disambiguation. The numbers beneath each figure denote the rotation (R) and translation (t) errors. Note that enhancing the intra-set distinctiveness consistently improves registration accuracy.

ambiguity, resulting in indistinguishable point features. Figure 1a demonstrates this phenomenon, henceforth referred to as the *indistinguishable feature problem*, which results in an overwhelming number of ambiguous point-to-point correspondences as opposed to the sharp matches desired for accurate registration.

To address these issues, we propose **D**istinctiveness **o**riented **P**ositional **E**quilibrium (*DoPE*), a novel, lightweight positional embedding module that significantly improves the intra-set distinctiveness of both the source and target point cloud embeddings, thereby enhancing the resulting point-to-point correspondences. Specifically, *DoPE* disambiguates the per-point features by augmenting them with global positional information computed with respect to the centroid of the combined source and target point clouds, which act as the origin of a shared coordinate system. We use this *DoPE* module in an iterative registration framework, where by the two point clouds are gradually aligned via rigid transformations that are computed from *DoPE*'s position-aware features. The joint origin and the correspondence matrix are alternately refined such that, the *DoPE* module feeds increasingly consistent positional information to would-be corresponding pairs and enhances the resulting correspondence predictions used to estimate the rigid transformation, which in turn updates the joint-origin used to provide positional embeddings in the next iteration. Within only a few iterations, the network converges to the so-called *positional equilibrium*, the desired fixed point with high registration accuracy where the positional embeddings given to matching pairs become essentially identical. In summary, the **contributions** are as follows:

- We identify and analyze the contributing factors to the *indistinguishable features problem*, a critical bottleneck to point-cloud registration that is prevalent in GNN-based architectures but has evaded scrutiny in the recent registration literature.
- To address this issue, we propose the **D**istinctiveness **o**riented **P**ositional **E**quilibrium (*DoPE*) module, a novel positional embedding scheme that disambiguates point features and enhances the resulting rigid-transformation predictions. We use *DoPE* as part of an iterative registration framework, whereby the two point clouds are gradually aligned by rigid transformations computed from *DoPE*'s position-aware features.
- We demonstrate the effectiveness of *DoPE* by incorporating the module into the state-of-the-art registration architectures and performing comprehensive experiments on various registration datasets and task settings, yielding *unprecedented performance improvement across all combinations*.

2. Related Work

Deep Learning on Point Clouds Deep Learning on point clouds was pioneered by PointNet [11], which directly consumes the input point clouds without any approximating transforms (*e.g*., voxelization or 2D projection) by embedding each input point independently using a shared multilayer perceptron (MLP). PointNet also achieves permutation invariance by aggregating the final features using a max-pooling layer as a symmetric function. However, the independent processing of the input points precludes PointNet from being able to capture local geometry in its features, a trait that has shown to be significant to point cloud registration [18].

Graph neural networks (GNN) provide a natural way to encode the local geometry of point clouds by virtue of local aggregation at each layer. For example, PointNet++ [12] recursively applies PointNet on a locally constructed graph (*e.g*., ball query or k-NN graphs); dynamic graph convolutional neural networks (DGCNN) [20] constructs a local neighborhood graph in the feature space and applies local feature aggregation on the edges connecting neighboring pairs of points. While GNN-based approaches have made significant improvements to 3D vision tasks such as point cloud classification and segmentation, many GNN backbones are prone to oversmoothing [6, 17, 1] and structural ambiguity, resulting in indistinguishable point features, which are detrimental to the feature matching step in point cloud registration.

Learning-based Registration The latest learning-based approaches to registration [18, 19, 24, 5, 25] have largely focused on improving the matching process between the embedded feature descriptors of the input point clouds, which are typically generated by a graph neural network (GNN). Specifically, DCP [18] finds matched correspondences from learned features via attention combined with pointer generation, while RPM-Net [24] incorporates Robust Point Matching [3] into a learning framework to be able to handle missing correspondences. PRNet [19] and IDAM [5] both extract keypoints and then iteratively find keypoint-to-keypoint correspondences. DeepGMR [25] explicitly proposes a probabilistic registration model by using Gaussian Mixture Model (GMM) parameters.

However, these methods remain oblivious to the fact that the GNN backbones typically used to embed the input point clouds are prone to the indistinguishable feature problem, thereby severely lacking the intra-set distinctiveness required to generate accurate point-to-point correspondences from the embedded features. In this paper, we identify and analyze the contributing factors to the indistinguishable features problem and propose a novel positional embedding module to significantly enhance the intra-set distinctiveness of the per-point features.

Concurrent work [8] has also suggested the use of positional encoding to improve the intra-set distinctiveness of the point descriptors. However, the positional encoding scheme in [8] remains local to each point cloud and was not born out of the full awareness of the underlying registration bottleneck inadvertently addressed by the proposed method. On the other hand, *DoPE*, fully motivated by the indistinguishable features problem, feeds positional embeddings computed with respect to the centroid of the combined source and target point clouds, which act as the origin of a *shared* coordinate system. As a result, the network remains aware of the relative spatial orientation of the point clouds during the iterative registration process, a trait that we empirically show to be indispensable to *DoPE*'s outstanding performance.

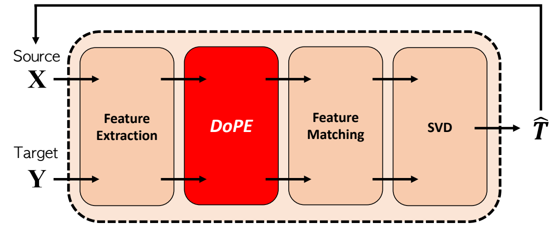


Figure 2: Overall flow for iterative registration process

3. Preliminaries

3.1. Problem Statement

Point cloud registration is the process of finding a rigid transformation that best aligns two unaligned point clouds. Let $\{\mathcal{X}, \mathcal{Y}\}$ be two finite point sets, which contain J and K points, respectively. Assuming that $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\} \subset \mathcal{X}$ and $\{\mathbf{y}_{\mathbf{x}_1}, \mathbf{y}_{\mathbf{x}_2}, \mathbf{y}_{\mathbf{x}_3}, ..., \mathbf{y}_{\mathbf{x}_N}\} \subset \mathcal{Y}$ are two sets of corresponding point clouds and N is the number of corresponding pairs ($N \leq J$ and $N \leq K$), the optimal rotation $\hat{\mathbf{R}}$ and translation $\hat{\mathbf{t}}$ are estimated as follows:

$$(\hat{\mathbf{R}}, \hat{\mathbf{t}}) = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3}{\arg\min} \sum_{i=1}^{N} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - \mathbf{y}_i\|^2, \quad (1)$$

where $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ comprise the rigid transformation that best aligns the two point clouds.

3.2. Iterative Registration Process

To find the optimal transformation in Eq. (1), many works [19, 24, 5] follow the iterative procedure shown in Figure 2. In each iteration, the source and target point clouds $\mathcal{X}$ and $\mathcal{Y}$ are first fed into the feature extraction layer to generate the high-level features. Next, the feature matching layer finds point-to-point correspondence. Finally, the optimal transformation $\hat{\mathbf{T}} = \begin{bmatrix}\hat{\mathbf{R}} & \hat{\mathbf{t}}; \mathbf{0} & 1\end{bmatrix}$ is estimated using Singular Value Decomposition (SVD) [14]. The whole process is repeated with $\mathcal{Y}$ and $\mathcal{X}$ transformed by $\hat{\mathbf{T}}$ computed in the previous iteration, until the estimated transformation converges to the ground-truth. In this work, we propose a lightweight, efficient module called *DoPE* and use it as part of this iterative registration framework in between the feature extraction and matching layers. By doing so, we enhance the intra-set distinctiveness of the intermediate features of every iteration and achieve increasingly more accurate point-to-point correspondences.

3.3. Feature Ambiguity in Learning-based Point Cloud Registration

In this sub-section, we identify and analyze the contributing factors to the above-mentioned *indistinguishable feature problem*. In doing so, we demonstrate that *lack* of intra-set distinctiveness has been a huge bottleneck to the latest GNN-based registration architectures and describe the motivations for *DoPE*'s design.

Indistinguishable features in GNNs are largely manifested in two ways: first, it has been empirically shown that a wide variety of GNN is prone to the *oversmoothing* problem [6, 17, 1], a phenomenon whereby repeated application of the message propagation step in each GNN layer renders all graph nodes to converge towards similar features across the *entire* point cloud, as displayed in Figure 1a. This is severely detrimental to the downstream registration procedure as it makes it challenging for the network to find the best match for a given point of point set $\mathcal{X}$ if there is very little difference amongst all of the point features of point set $\mathcal{Y}$, and vice versa. The oversmoothing problem in GNNs can be alleviated to an extent by attention mechanisms [10, 9] thanks to their data-dependent, attention-weighted aggregation scheme, with a similar argument having been made for oversmoothing in CNNs for images [27, 21]. This is corroborated in Figure 1, where the features of the backbone processed by both local (Figure 1b) and non-local (Figure 1c) attention are significantly more distinctive than the raw vanilla features (Figure 1a). More notably, the increase in intra-set distinctiveness is accompanied by a meaningful improvement to the registration error.

However, attention-based feature aggregation fails to address what is a more subtle contributing factor to indistinguishable GNN features: *structural ambiguity*, a problem induced by the (partial) translation invariance encoded in the message propagation step of prominent GNN backbones [18, 19, 24, 5] that renders points that are separate, but encode locally similar structures of the point cloud to have near-identical features. For example, the point feature enclosed by the red circle in Figure 1b and 1c remain similar to the corresponding points that lie on the other legs of the chair. Such features can potentially hamper the registration process due to spurious matches between locally similar structures in the source and target point clouds. This is where the positional embedding in the proposed *DoPE* module comes into the equation; it provides the network with additional cues to distinguish between the four legs and to be able to appropriately match the constituent points of each leg to the correct counterpart in the target point cloud. As shown in Figure 1d and 1e, the positional embedding further enhances the intra-set distinctiveness to its upper limit, yielding significant improvements to registration accuracy. This demonstrates that although structural ambiguity isn't as conspicuous of a phenomenon as is oversmoothing, it has been the biggest bottleneck to the latest GNN-based registration architectures, an observation that has not only motivated our work but one that will hopefully motivate the future design of GNNs for registration.

4. Proposed Method

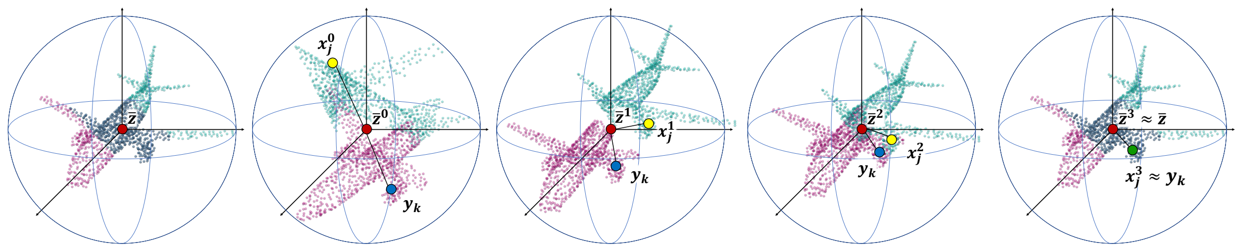
We now present *DoPE*, a novel positional embedding unit used as part of an iterative registration framework (Figure 2) that disambiguates the backbone GNN features for more effective point cloud registration. In Section 4.1, we introduce the constituent operations of the *DoPE module* and its properties; in Section 4.2, we describe how the iterative application of the *DoPE* module converges the network to the positional equilibrium, a fixed point with high registration accuracy where matching points are given essentially identical positional embeddings; finally, in Section 4.3, we outline the loss function that we use to further encourage feature disambiguation in an end-to-end manner.

4.1. DoPE Module

Joint-origin Update The *DoPE* module disambiguates the backbone features via positional embeddings followed by a non-local attention operation. In order to feed positional embeddings to both the source and target point clouds $\mathcal{X}$ and $\mathcal{Y}$, we first compute the *joint origin*, the origin of a shared coordinate system where the positional information is defined. In iteration t of the forward pass of the registration pipeline, we update the joint-origin $\bar{\mathbf{z}}^{(t)}$ as the center-of-mass of the union of $\mathcal{X}^{(t)} = \{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \cdots, \mathbf{x}_J^{(t)}\}$ and $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K\}$:

$$\bar{\mathbf{z}}^{(t)} = \frac{1}{J+K}\left(\sum_{i=1}^{J}\mathbf{x}_i^{(t)} + \sum_{i=1}^{K}\mathbf{y}_i\right), \tag{2}$$

The joint-origin is an essential aspect of the *DoPE* module. Computing positional embeddings with respect to a shared coordinate frame allows the network to remain aware of the relative spatial orientation of the point clouds during iterative registration. Furthermore, updating the joint-origin as the centroid of the combined point clouds enforces a special type of translation invariance, whereby the *DoPE* module feeds the same set of positional embeddings to point cloud pairs with the same relative configuration in 3D space, but located in different absolute positions. As a result, our registration architecture is invariant to such variations that may occur within the dataset itself or even throughout the iterative process, thereby narrowing down the space of regis-



(a) ***Ground truth*** (b) ***Initial input*** (c) ***After 1st iteration*** (d) ***After 2nd iteration*** (e) ***After 3rd iteration***

Figure 3: The update process of joint-origin $\bar{\mathbf{z}}^{(t)}$ according to the registration iteration. The positional information from the joint-origin of two matching points $\mathbf{x}_j^{(t)}$ and $\mathbf{y}_k$ become increasingly identical.

tration scenarios faced by the *DoPE* module and benefiting the overall training procedure.

Feature Disambiguation To disambiguate the backbone features, the positional equilibrium explicitly combines the positional information with respect to the joint-origin with the backbone high-level features via self-attention. After estimating joint origin $\bar{\mathbf{z}}^{(t)}$ after t-th iteration, the positional equilibrium first computes the positional embedding of each point, and then add to the backbone features as follows:

$$\begin{aligned} F_{\mathbf{x}_i} &\leftarrow F_{\mathbf{x}_i} + \mathcal{M}(\mathbf{x}_i^{(t)} - \bar{\mathbf{z}}^{(t)}), \\ F_{\mathbf{y}_i} &\leftarrow F_{\mathbf{y}_i} + \mathcal{M}(\mathbf{y}_i - \bar{\mathbf{z}}^{(t)}), \end{aligned} \tag{3}$$

where $\mathcal{M}$ is a shared multi-layer perceptron (MLP). In addition to adding the positional embedding into the backbone features, aggregating other contextual cues via self-attention [16] can intuitively increase the distinctiveness of each point as follows:

$$F_{\mathbf{x}_i} \leftarrow \sum_{j \in \mathcal{S}(X)} \alpha_{\mathbf{x}_{ij}} F_{\mathbf{x}_j}, \quad F_{\mathbf{y}_i} \leftarrow \sum_{j \in \mathcal{S}(Y)} \alpha_{\mathbf{y}_{ij}} F_{\mathbf{y}_j}, \tag{4}$$

where $\alpha_{\mathbf{x}_{ij}} = \text{Softmax}_j({\mathbf{q}_{\mathbf{x}_i}}^T \mathbf{k}_{\mathbf{x}_j})$ is the similarity between i-th query and j-th key features of $\mathbf{x}$ by setting $\mathbf{q}_{\mathbf{x}_i} = F_{\mathbf{x}_i}$ and $\mathbf{k}_{\mathbf{x}_j} = F_{\mathbf{x}_j}$. $\alpha_{\mathbf{y}_{ij}}$ is also defined as identical to $\alpha_{\mathbf{x}_{ij}}$. Because the feature update process in Eq. (4) has quadratic space-time complexity when the features are aggregated from all points, we sample the features to be aggregated (*i.e.* we sample the keys) to model the long-range dependencies with light-weight memory and computation. Inspired by [4], we use random sampling to uniformly select $\|\mathcal{S}\|$ points from which the features are aggregated. Since $N \gg \|\mathcal{S}\|$, the complexity of *DoPE* is low with the order of $O(N \cdot \|\mathcal{S}\|)$.

4.2. Positional Equilibrium

In the iterative registration framework outlined in Figure 2, the joint origin and the correspondence matrix are alternately refined such that, the *DoPE* module feeds increasingly consistent positional information to would-be corresponding pairs and enhances the resulting correspondence predictions. In turn, the rigid transformation estimated from this correspondence matrix updates the joint-origin used to provide positional embeddings in the next iteration. For example, in Figure 3, we conduct point cloud registration on source (blue-green) and target (pink) point clouds of the airplane. Assume that $\mathbf{x}_j$ and $\mathbf{y}_k$ are the corresponding pair of the source and target points located in the engine part of the airplane. We denote $\mathbf{x}_j^0$ as the initial point of $\mathbf{x}_j$. At the beginning of the registration process (Figure 3b,) the positional information of $\mathbf{x}_j^0$ and $\mathbf{y}_k$ is not so close to each other, leading to the mismatch in Figure 3c. Because the positional information of $\mathbf{x}_j^0$ and $\mathbf{y}_k$ is closer, the network predicts more correct correspondence matrix (Figure 3d.) Within only a few iterations, $\mathbf{x}_j^{(t)}$ converges to $\mathbf{y}_k$, indicative of the *positional equilibrium* where the positional embeddings given to matching pairs become essentially identical, as shown in Figure 3e.

4.3. Loss Function

To encourage the network to learn distinctive feature descriptors, we adopt a loss function based on the equilibrium-state correspondence matrix of our architecture. For the ideally distinctive feature descriptors, the feature descriptors should have high similarity between matching pairs and should have low similarity between non-matching pairs. Let assume that the correspondence matrix outputted from the feature matching layer is as follows:

$$\mathbf{P} = \{p_{jk}\}^{J \times K}, \quad 0 \le p_{jk} \le 1 \tag{5}$$

Without loss of generality, p_{jk} is scaled similarity between $F_{\mathbf{x}_j}$ and $F_{\mathbf{y}_k}$. Because each element of the equilibrium-state correspondence matrix, p_{jk}, represents zero for all j and k except that x_j and y_k are matching pair (*i.e.* $p_{jk} = 1$ if $\mathbf{x}_j$ and $\mathbf{y}_k$ are matching pair and $p_{jk} = 0$ if not) which are identical to the elements of ground-truth correspondence matrix, we thus supervise our network to learn ground-truth correspondences as:

$$\mathcal{L}_{corr} = -\sum_{j=1}^{J} log(p_{jk^*}), \tag{6}$$

where $\mathbf{y}_{k^*}$ is the ground truth target point corresponded with source point $\mathbf{x}_j$. The correspondence loss, $\mathcal{L}_{corr}$, is a cross-entropy loss used in [5, 2]. We employ this loss specifically because the correspondence loss further strengthens the distinctiveness of feature descriptors between non-matching pairs and the matchability between matching pairs ($\because \sum_{k=1}^{K} p_{jk} = 1$.)

The correspondence loss is prone to the overfitting problem because it guides the point-to-point correspondence about all matching pairs. To alleviate the overfitting, we additionally add the transformation loss $\mathcal{L}_{trans}$ to the total loss of network as the regularization term as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{corr} + \lambda \mathcal{L}_{trans}, \tag{7}$$

where $\mathcal{L}_{trans} = \|\hat{\mathbf{R}}^T \mathbf{R}^* - \mathbf{I}\|_2^2 + \|\hat{\mathbf{t}} - \mathbf{t}^*\|_2^2$ is also used by existing methods [18, 19, 24, 2, 25]. As mentioned in Section 3.2, we fuse our *DoPE* into the other registration methods and compute the total loss at every t-th iteration.

5. Experiments

For following sections, we evaluate the performance of our *DoPE* by inserting it into the various baseline registration methods with the various datsets. The learning-based registration methods such as Deep Closest Point (DCP [18]), PRNet [19], RPM-Net [24], Iterative Distance-Aware Matrix convolution (IDAM [5]), DeepGMR [25], and Deep Global Registration (DGR [2]) are considered as our baseline networks. The object-level datasets (ModelNet40 [22], ScanObjectNN [15]) and the scene-level dataset (3DMatch [26]) are employed. We reported the superior result between the original paper and our reruns for the results of baseline methods.

5.1. Object-level Dataset

The ModelNet40 dataset consists of $12,311$ models from 40 categories. As with the DCP [18], we divide $12,311$ models into $9,843$ for training and $2,468$ for testing. During training, we pick rotation R and translation t randomly at $[0, 45°]$ and $[-0.5, 0.5]$, respectively. Then, we measure the root-mean-square error (RMSE), and the mean absolute error (MAE) between the ground-truth $(\mathbf{R}^*, \mathbf{t}^*)$, and the predicted $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$. The rotation measurements is the degree. We use DCP, PRNet, IDAM, RPM-Net as our baseline methods of the object-level dataset.

Full Data Full data setting implies that source and target point clouds have exact one-to-one correspondences for all points. Specifically, we follow the experimental settings of DCP [18] for the full data of ModelNet40, sampling $1,024$ points from the surface of each model of ModelNet40. Table 1 shows the registration results on full data without any perturbation of points (full+clean dataset.) *DoPE* remarkably enhances the performance of baseline methods. RPM-Net+*DoPE* especially achieves dozens of times performance improvement in terms of rotational metric compare to baseline performance. Moreover, we also investigate the robustness to Gaussian noise in Table 2. We add random Gaussian noise with the distribution of $\mathcal{N}(0, 0.01)$ to each point of source and target point clouds independently so that some points could not have exact matching points. Table 2 shows that *DoPE* still noticeably improves the registration performance across all benchmarks in the presence of noise, although the positional information of matching points could not be identical due to the noise.

Models	RMSE(R)↓	MAE(R)↓	RMSE(t)↓	MAE(t)↓
DCP [18]	1.143385	0.770573	0.001786	0.001195
DCP+*DoPE*	**0.383430**	**0.085278**	**0.001224**	**0.000512**
PRNet [19]	2.1425	0.960	0.00943	0.006
PRNet+*DoPE*	**0.264531**	**0.158885**	**0.003669**	**0.002054**
IDAM [5]	1.556997	1.014915	0.019774	0.012126
IDAM+*DoPE*	**0.543097**	**0.309420**	**0.003495**	**0.002103**
RPMNet [24]	0.084	0.028	0.00032	0.00016
RPMNet+*DoPE*	**0.0017**	**0.0009**	**0.00003**	**0.00003**
DeepGMR [25]	0.0125	0.0008	0.0001	0.0000
DeepGMR+*DoPE*	**0.003**	**0.0002**	**0.0000**	**0.0000**

Table 1: Results on ModelNet40 full+clean dataset

Models	RMSE(R)↓	MAE(R)↓	RMSE(t)↓	MAE(t)↓
DCP [18]	7.224	4.528	0.0514	0.0345
DCP+*DoPE*	**3.4770**	**1.6240**	**0.0071**	**0.004396**
PRNet [19]	2.755183	1.219011	0.010428	0.007927
PRNet+*DoPE*	**0.615637**	**0.425077**	**0.006396**	**0.004585**
IDAM [5]	2.3917	0.8335	0.008760	0.004363
IDAM+*DoPE*	**0.900124**	**0.564658**	**0.005179**	**0.003647**
RPMNet [24]	1.1587	0.343	0.0068	0.0030
RPMNet+*DoPE*	**0.1057**	**0.0795**	**0.001011**	**0.0007823**
DeepGMR [25]	1.75319	1.00646	0.00485	0.002849
DeepGMR+*DoPE*	**0.9205**	**0.6073**	**0.002959**	**0.001998**

Table 2: Results on ModelNet40 full+noisy dataset

Partial Data Because point cloud registration mostly occurs between partially overlapped point clouds in real-world applications, we generate the partially overlapped data of ModelNet40. We randomly pick one point from each source and target point clouds and then compute 768 nearest-neighbor points out of the full $1,024$ points as in PRNet [19]. Table 3 shows the results of partial+clean ModelNet40. *DoPE* enhances all baseline's performance even better than the existing SOTA performance. Especially, DCP with *DoPE* surprisingly outperforms other baseline methods about the all performance metrics, although DCP does not explicitly handle the partially-overlapped registration problem. This indicates that the distinctiveness of features is significant for point cloud registration. We also experiment with Gaussian noise (partial+noisy dataset) and show that our proposed module is also robust to the noise in Table 4.

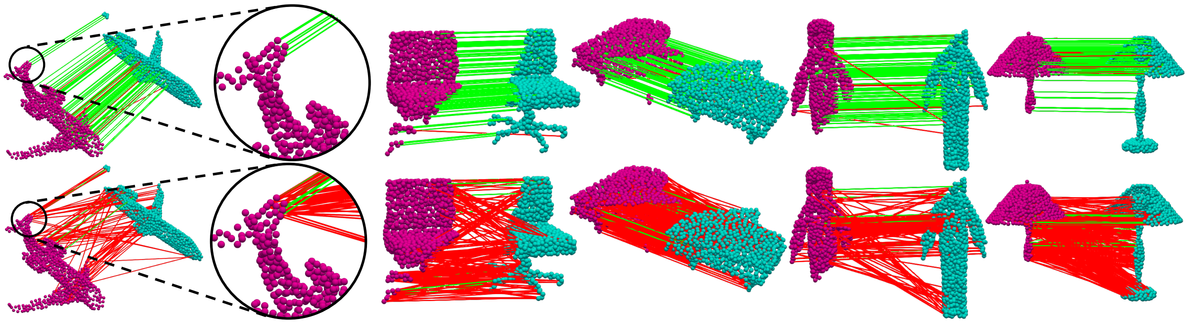


Figure 4: Qualitative comparison of matching pairs between ***DCP+DoPE*** (upper) and ***DCP*** (lower). Green line indicates the correct matching pair whereas red line indicates the incorrect matching pair.

Models	RMSE(R)↓	MAE(R)↓	RMSE(t)↓	MAE(t)↓
DCP [18]	6.709	4.448	0.027	0.020
DCP+*DoPE*	**1.9125**	**0.800102**	**0.011467**	**0.006442**
PRNet [19]	3.1993	1.454	0.016	0.010
PRNet+*DoPE*	**0.514125**	**0.311634**	**0.008571**	**0.005826**
IDAM [5]	2.95	0.76	0.021	0.005
IDAM+*DoPE*	**0.405834**	**0.307274**	**0.002063**	**0.001598**
RPMNet [24]	2.4961	0.8428	0.01919	0.007871
RPMNet+*DoPE*	**0.1944**	**0.1178**	**0.001889**	**0.001250**

Table 3: Results on ModelNet40 partial+clean dataset

Models	RMSE(R)↓	MAE(R)↓	RMSE(t)↓	MAE(t)↓
DCP [18]	7.884786	5.834028	0.040586	0.030548
DCP+*DoPE*	**5.8885**	**3.1487**	**0.0170**	**0.0104**
PRNet [19]	4.323	2.051	0.017	0.012
PRNet+*DoPE*	**0.614777**	**0.406517**	**0.009008**	**0.006338**
IDAM [5]	3.72	1.85	0.023	0.011
IDAM+*DoPE*	**0.560017**	**0.423182**	**0.003644**	**0.002962**
RPMNet [24]	2.8818	1.0038	0.02054	0.008715
RPMNet+*DoPE*	**0.2143**	**0.1566**	**0.001917**	**0.001437**

Table 4: Results on ModelNet40 partial+noisy dataset

Unfamiliar Data To compare each method's generalizability, we test on ScanObjectNN dataset using the models that are trained on full+noisy ModelNet40 dataset in Table 4. Because the ScanObjectNN is a real-world point cloud object dataset extracted from scanned indoor scene data consisting of $15,000$ objects categorized into 15 categories, ScanObjectNN contains objects which are significantly different from the synthetic CAD dataset-ModelNet40. Because the ScanObjectNN data are extracted from scanned scene data, background elements or parts of nearby objects could be included in each object data, and even the density of point clouds is non-uniform. The results on ScanObjectNN is shown in Table 5, indicating that the *DoPE* module works well even in the unfamiliar data.

5.2. Scene-level Dataset

For the scene-level dataset, we use the real-world indoor 3DMatch dataset [26], which consists of 3D point cloud pairs from eight different scenes with ground truth transformations estimated from RGB-D reconstruction and use DGR as our baseline. A single point is subsampled within each 5cm voxel to generate point clouds with uniform density. We follow the train/test split and the standard procedure to generate pairs with at least 30% overlap for training and testing. Different from the error metrics in synthetic dataset, we use the error metric as DGR does for fair comparison: rotation error (RE) as $arccos\frac{Tr(\hat{\mathbf{R}}^T\mathbf{R})-1}{2}$, translational error (TE) as $\|\hat{\mathbf{t}}-\mathbf{t}\|_2^2$, and recall. Recall is the ratio of successful registrations, and we define a successful registration as the case in which RE is less than 15 degrees, and TE is less than 0.3m. Table 6 summarizes the experimental results of the 3DMatch dataset. DGR+DoPE outperforms the baseline significantly, demonstrating that DoPE can be scalable well in the scene-level dataset.

Models	RMSE(R)↓	MAE(R)↓	RMSE(t)↓	MAE(t)↓
DCP [18]	7.923467	5.749913	0.025723	0.020359
DCP+*DoPE*	**4.556273**	**1.731377**	**0.009549**	**0.004982**
PRNet [19]	1.4486	0.711052	0.0066	0.0050
PRNet+*DoPE*	**0.271568**	**0.201960**	**0.004669**	**0.003531**
IDAM [5]	2.3718	1.2968	0.010578	0.005594
IDAM+*DoPE*	**0.9063**	**0.5737**	**0.0058**	**0.0037**
DeepGMR [25]	2.79	1.223	0.0085	0.0061
DeepGMR+*DoPE*	**1.48**	**0.7713**	**0.0057**	**0.0038**

Table 5: Results on ScanObjectNN dataset

Models	Recall↑	TE(cm)↓	RE(deg)↓
FGR [28]	42.70%	10.60	4.08
RANSAC-2M [13]	66.10%	8.85	3.00
RANSAC-4M [13]	70.70%	9.16	2.95
RANSAC-8M [13]	74.90%	8.96	2.92
Go-ICP [23]	22.90%	14.70	5.38
Super4PCS [7]	21.60%	14.10	5.25
ICP (P2Point) [29]	6.04%	18.10	8.25
ICP (P2Plane) [29]	6.59%	15.20	6.61
DGR [2]	91.3%	7.34	2.43
DGR+*DoPE*	**96.6%**	**6.09**	**1.63**

Table 6: Results on 3DMatch dataset

GNN(DCP)	LA	N-LA	Indiv.	Joint	MAE(R)↓	MAE(t)↓
✓					5.34	0.022
✓	✓				4.74	0.022
✓	✓		✓		2.38	0.021
✓	✓			✓	1.04	**0.006**
✓		✓			4.45	0.020
✓		✓	✓		3.86	0.025
✓		✓		✓	**0.80**	**0.006**

Table 7: Effects of each component on registration performance. **LA**: Local attention, **N-LA**: Non-local attention, **Indiv.**: Positional embeddings computed w.r.t. individual origin (centroid of each point cloud), **Joint**: Positional embeddings computed w.r.t. joint-origin

6. Analysis

6.1. Ablation Study

Effects of Different Components Table 7 shows that the performance improvement induced by *DoPE* is largely attributed to the use of positional embeddings *w.r.t. the joint origin* rather than mere vanilla adoptions of existing operations such as self-attention and positional embeddings. Table 7 demonstrates that the use of the joint-origin is not arbitrary and is in fact an crucial aspect of *DoPE*'s design.

Models	$\mathcal{L}_{corr}$	MAE(R)↓	MAE(t)↓
DCP [18]	No	**4.45**	**0.02**
	Yes	5.61	0.03
DCP+*DoPE*	No	1.33	0.009
	Yes	**0.80**	**0.006**
PRNet [19]	No	**1.45**	**0.010**
	Yes	1.72	0.018
PRNet+*DoPE*	No	0.31	0.006
	Yes	**0.20**	**0.002**
RPMNet [24]	No	**0.84**	**0.008**
	Yes	1.41	0.012
RPMNet+*DoPE*	No	0.14	0.0013
	Yes	**0.12**	**0.0011**

Table 8: Effects of $\mathcal{L}_{corr}$

Effects of $\mathcal{L}_{corr}$ Table 8 shows that the increase in registration accuracy is mainly attributed to *DoPE* as opposed to the correspondence loss $\mathcal{L}_{corr}$. Simply adding the correspondence loss to the baselines actually *hampers* performance, whereas the *DoPE* module improves registration accuracy by enhancing the point-wise correspondences and thereby enabling the backbone network to better leverage the correspondence loss.

6.2. Visualization Analysis

In Figure 4, we visualize influence of the *DoPE* module through its point-to-point correspondence matrix. Specifically, we plot the line connecting points $\mathbf{x}_j$ and $\mathbf{y}_k$ of source and target point clouds, respectively, when the matching score between the two points exceeds 0.01 (*i.e.*, if $p_{jk} > 0.01$). The line is colored green if the correspondence is correct and red if it is incorrect. By comparing the correspondence matrices of **DCP** and **DCP+*DoPE*** for several ModelNet40 objects, we see that DCP+*DoPE* predicts more accurate matching matrices than does DCP for *all* objects.

Models	# of params	# of FLOPs	Inference time(ms)
DCP [18]	5.60M	29B	18
DCP+*DoPE*	7.80M	33B	21
PRNet [19]	5.70M	90B	60
PRNet+*DoPE*	8.00M	99B	68
IDAM [5]	0.09M	1B	0.4
IDAM+*DoPE*	0.12M	1.3B	0.5
RPMNet [24]	1.82M	61B	36
RPMNet+*DoPE*	2.13M	78B	46
DeepGMR [25]	1.64M	5.8B	3
DeepGMR+*DoPE*	1.84M	6.4B	4
DGR [2]	243.87M	2,796B	2,100
DGR+*DoPE*	244.11M	2,802B	2,200

Table 9: Efficiency on various methods

The leftmost object in Figure 4 shows in greater detail that the lines on the horizontal stabilizer represent exact one-to-one matching between source and target point clouds in DCP+*DoPE* whereas the lines in DCP represent ambiguous and incorrect matching. Through these qualitative visualization results, we further illustrate the problems induced by oversmoothing and structural ambiguity of the GNN backbone, and how *DoPE* alleviates their effects.

6.3. Efficiency Analysis

We estimate the efficiency of various models using the number of network parameters, the number of FLOPs, and the inference time. The FLOPs and inference time are estimated for processing one pair of input point clouds. We use the same hyper-parameter settings as reported by each method. Table 9 demonstrates that the the *DoPE* module incurs little additional complexity compared to the baseline models, indicating the potential scaleability of *DoPE* to models that handle large data settings.

7. Conclusion

In this paper, we call to attention the shortcomings of GNN-based features for registration, namely the indistinguishable feature problem associated with oversmoothing and structural ambiguity. These constituent issues motivate *DoPE*, a novel positional embedding module that significantly enhances the intra-set distinctiveness of the per-point features generated by prominent GNN backbones, and hence the resulting point-to-point correspondences. *DoPE* computes positional information with respect to the joint-origin of the combined point clouds, iteratively refining the joint-origin and the correspondence matrix until convergence to an equilibrium where the positional embedding for both point clouds become essentially identical. We demonstrate that the *DoPE* module significantly increases the registration performance across all combinations of experimental settings. Furthermore, we hope that our analysis of the indistinguishable features problem motivates the future design of a stand-alone GNN backbone specifically tailored to point cloud registration.

References

[1] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[2] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition (PR)*, 31(8):1019–1031, 1998.

[4] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *European Conference on Computer Vision (ECCV)*, 2020.

[6] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[7] Nicolas Mellado, Dror Aiger, and Niloy J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. *Computer Graphics Forum (CGF)*, 33(5):205–215, 2014.

[8] Taewon Min, Eunseok Kim, and Inwook Shim. Geometry guided network for point cloud registration. *IEEE Robotics and Automation Letters (RA-L)*, 6(4):7270–7277, 2021.

[9] Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering gcn: Overcoming oversmoothness in graph convolutional networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] Hesham Mostafa and Marcel Nassar. Permutohedral-gcn: Graph convolutional networks with global attention. *arXiv preprint arXiv:2003.00635*, 2020.

[11] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[13] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[14] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1(1):1–5, 2017.

[15] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[17] Guangtao Wang, Rex Ying, Jing Huang, and Leskovec Jure. Improving graph attention networks with large margin-based constraints. *arXiv preprint arXiv:1910.11945*, 2019.

[18] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[19] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[20] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019.

[21] Zhihui Wang, Shijie Wang, Shuhui Yang, Haojie Li, Jianjun Li, and Zezhou Li. Weakly supervised fine-grained image classification via gaussian mixture model oriented discriminative learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[22] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[23] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(11):2241–2254, 2015.

[24] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *European Conference on Computer Vision (ECCV)*, 2020.

[26] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *International Conference on Learned Representations (ICLR)*, 2019.

[28] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision (ECCV)*, 2016.

[29] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.