# TopFormer: Topology-Aware Transformer for Point Cloud Registration

Sheldon Fung[1], Wei Pan[2], Xiao Liu[3], John Yearwood[3], Richard Dazeley[3], and Xuequan Lu[1]

[1] La Trobe University, Australia
[2] OPT Machine Vision, China
[3] Deakin University, Australia

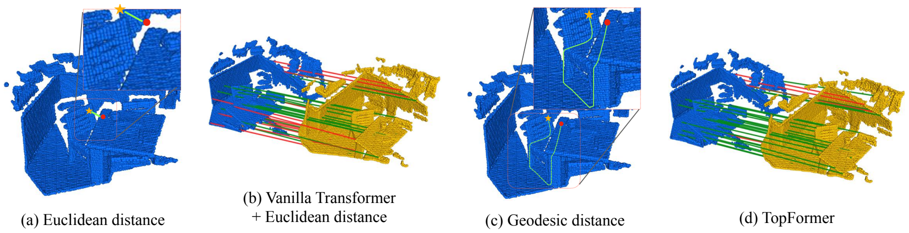
**Abstract.** The extraction of robust feature descriptors is crucial for achieving accurate point cloud registration. While the attention mechanism plays an important role in enabling sparse point features to learn global position-aware contextual information, the high sparsity at sub-sampled points can yield ambiguity in the corresponding features due to the loss of fine-grained structural information. In this paper, we propose *TopFormer*, a topology-aware Transformer that leverages surface-based geodesic topology to learn robust feature descriptors for point cloud registration. In particular, we design a topological structure encoding to capture point-pair surface-based structure in a sparse-through-dense manner. It couples the geodesic distance with the normal-based directional information, which provides a strong topological relation between each point pair. The proposed sparse-through-dense strategy is achieved by querying the information (e.g., geodesic distance) calculated from the dense point cloud for a pair of sparse points that exist in the dense point cloud. By doing so, the Transformer is able to learn feature descriptors that are more aware of the surface-based structural information. We evaluate the performance of our method on both indoor and outdoor datasets with different point cloud pair overlapping ratios. Experimental results show that our approach produces higher registration recalls than state-of-the-art techniques.

**Keywords:** Geodesic topology · point cloud registration

## 1 Introduction

Given a pair of partially overlapped point clouds, point cloud registration seeks to estimate the relative posture and further predict the transformation to align them together. This is a long-existing and fundamental task in the visual media and 3D vision field [23, 24], which can facilitate a wide range of practical scenarios such as autopilot, VR/AR applications, and automated robot positioning, among others. As 3D imaging devices such as RGB-D cameras, structured light scanners, and LiDAR become more commercially available, point cloud registration has gained noticeable attention from researchers recently [7, 2, 17].

While the attention mechanism [33] originated from natural language processing [28], it has witnessed huge successes in various multimedia and vision tasks [5, 22, 6,



(a) Euclidean distance

(b) Vanilla Transformer
+ Euclidean distance

(c) Geodesic distance

(d) TopFormer

**Fig. 1.** Consider two points located at the top of a chair (the yellow star) and the corner of a desk (the red dot), respectively. As illustrated by the green lines (the shortest paths) in (a) and (c), they are close to each other in Euclidean space, and meanwhile are distant from each other along the surface. These characteristics are straightforward in dense points, which, however, is difficult to observe in sparse points. Our TopFormer proposes the sparse-through-dense encoding to capture surface-based topological structure that helps mitigate the correspondence error (see (b) and (d), where red lines represent the wrong matches).

18]. Recent research [16, 35] has adopted the attention mechanism in the point cloud registration task, resolving the matching ambiguity brought by the limited perception field of convolution-based neural networks, e.g., Kernel Point Convolution (KPConv) [32]. While the architecture of the Transformer allows for the interactions between each sparse point and all other sparse points within a given point cloud pair, it tends to overlook the ordering and position of those points. Recent works overcome such issues by either fusing the point-wise position to the sparse point features [41] or injecting point-pair relative position information into the Transformer [21, 27] to mitigate the potential ambiguity caused by the lack of non-local geometry information. Nevertheless, ambiguity among features might still arise under circumstances where two sparse points are spatially close but geodesically distant. As illustrated in Figure 1, considering two points located on a chair and a desk corner, simply merging relative position information to the local feature fails to overcome the feature ambiguity due to two reasons. 1) They share similar local geometry structures since they lie in the vicinity of one another. 2) The inserted relative position information fails to provide distinguishable information for them since they are close to each other in Cartesian space.

Human vision tends to perceive a point cloud as multiple surfaces and can further deduce the underlying topology (i.e., neighborhood relationships among points). By determining this, humans can easily identify the ambiguity of geometrically close but semantically faraway points. Motivated by this, we propose *TopFormer*, a novel point cloud registration approach that captures the fine-grained surface-based geodesic topology information. In particular, we design a topology structure encoding to fuse the surface-based geodesic distance with the normal-based directional information in a sparse-through-dense manner. More specifically, given a pair of sparse points that exist in the dense points, we obtain their geodesic distance by querying this information from dense points. It provides strong cues for learning discriminative features, especially for sparse points that are spatially close while geodesically distant. Then, we strengthen the encoding by fusing the geodesic distance with the normal-based directional information. The designed directional information overcomes the issue of normal orientation due to arbitrary poses since it is invariant against unoriented normals, as illustrated in Figure 2. Finally, we aggregate the encoding with the sparse local-based geometry features extracted from KPConv and adopt self-attention layers to learn global contextual features that are aware of topological structural information.

We evaluate the proposed TopFormer on 3DMatch [43] and KITTI [12] datasets. Extensive experiments show that our TopFormer outperforms state-of-the-art methods, demonstrating the effectiveness of our proposed method. Our contributions are:

- We propose TopFormer, which is capable of learning topology-aware structural features, yielding robust correspondence predictions.
- We design a topology structure encoding and a sparse-through-dense strategy to capture the pair-wise surface-based information.
- We conduct extensive experiments on the proposed method and compare the results with the state-of-the-art approaches, demonstrating the effectiveness of our method.

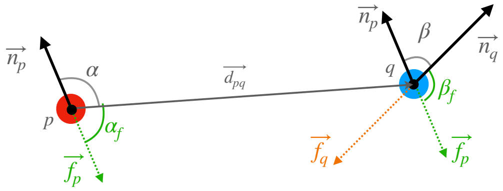


**Fig. 2.** An illustration of the invariance against unoriented normals. Given two sparse points $p$ and $q$ and their oriented normals $\vec{n_p}$ and $\vec{n_q}$, we can obtain the corresponding angle $\alpha$ and $\beta$. The unoriented normals might point in the opposite direction, represented by $\vec{f_p}$ and $\vec{f_q}$. Our proposed topological structure encoding ensures the invariance against unoriented normals since $\alpha + \alpha_f = \beta + \beta_f = \pi$ and therefore $\sin(\alpha) = \sin(\alpha_f) = \sin(\beta) = \sin(\beta_f)$.

## 2 Related Work

**Local feature descriptors.** In general, most point cloud registration methods follow a four-step pipeline: 1) extract feature descriptors, 2) form correspondences based on the similarity of the features, 3) reject the correspondences outliers (e.g., RANSAC), and 4) compute the transformation matrix with Singular Value Decomposition (SVD). Extensive research has been focused on the first yet the most crucial step. Zeng *et al.* [43] used a 3D convolution network to extract local feature descriptors from the truncated volumetric grid. Inspired by it, Gojcic *et al.* [13] leveraged the local reference frame (LRF) and proposed the smoothed density value (SDV), enabling sensor modalities generalization. Although pioneer works show promising results, they have limitations as they only consider local structural information of randomly sampled blocks. This makes it difficult to strike a balance between computational efficiency and repeatability. To overcome this, Deng *et al.* [8] built the local geometry representation by exploiting the hand-crafted Point Pair Feature (PPF) [10] and then used PointNet [26] to learn local features that are aware of the global context. Bai *et al.* [3] took an alternative approach and resorted to KPConv [32] to directly consume the unstructured point cloud. Then the extracted point features are supervised by a descriptor loss and detector loss in a joint learning fashion. Despite the high performance across different datasets, this method suffers from 1) limited perception fields of the convolution operation, and 2) the lack of awareness of the information from the other point cloud fragment. A milestone work, Predator [16], addressed the above-mentioned limitations by incorporating the graph neural network (GNN) [37] and Transformer [33] to respectively strengthen contextual information and allow mutual information exchange.

Transformers have witnessed significant breakthroughs in many fields including natural language processing [28] and 2D/3D visual data processing [25, 44], among others. Early attempts for adopting Transformers to point cloud registration usually overlook the position/structural information of the point cloud and instead feed the high-level point features directly into the Transformer [35, 42]. To fill the information gap between the unordered point cloud and Transformer, some recent works concentrated on combining the high-level point features input with structural-revealing embeddings. Li *et al.* [21] proposed to insert relative positional encoding in sparse point features to mitigate the ambiguity of the repetitive geometry patterns in the point cloud scene data. Likewise, Qin *et al.* [27] introduced the non-local geometric structure encoding, which strengthens the geometrical discriminability of the learned sparse point features, and on the other hand, provides transformation-invariant structural information for extracting robust features from point clouds with arbitrary poses. Yet, sparse points themselves are topologically less representative, leading to ambiguity in feature space, particularly for the spatially close but geodesically distant points. Different from the above methods, our insight is to design a topology structure encoding aiming to mitigate such ambiguity.

**End-to-end registration.** Another line of research is to estimate the transformation in an end-to-end manner. Following the pioneering work Iterative Closest Point (ICP) [4], researchers focus on employing deep learning techniques to predict soft correspondences iteratively, and the transformation parameters can be obtained by solving a weighted-SVD on the soft correspondences [35, 36, 40]. They effectively alleviate the issue of falling into local optima in the ICP method. Xu *et al.* [38] adopted the overlapping mask in the iterative process and aggregated the global feature with a max-pooling operation. Although it achieves nice performance in partial object data registration tasks, it yields less satisfactory results when applied to scene data. Yew *et al.* [41] took an alternative approach and directly predicted the final set of correspondences in the sparse point level by exploiting the powerful attention mechanism.

## 3 Methodology

Our approach adopts a coarse-to-fine pipeline inspired by established methods such as CoFiNet [42] and GeoTransformer [27]. The method unfolds in a structured sequence comprising five key steps: local feature extraction, global contextual feature extraction, sparse points matching, dense points refinement, and RANSAC outliers removal. Local geometry features are first extracted from the point clouds with KPConv. We then develop a topological structure encoding in a sparse-through-dense manner with the points in different sparsities outputted by KPConv. The encoding is further packed with local features and fed into the Transformer to aggregate global context while enabling mutual information exchange between two point clouds. We leverage the sparse points matching module to generate loose correspondences, which are subsequently used as constraints to obtain dense-level correspondences. Finally, RANSAC is applied to the dense-level correspondences, and the transformation that aligns two point clouds can be computed. Figure 3 illustrates the overview of our proposed method.

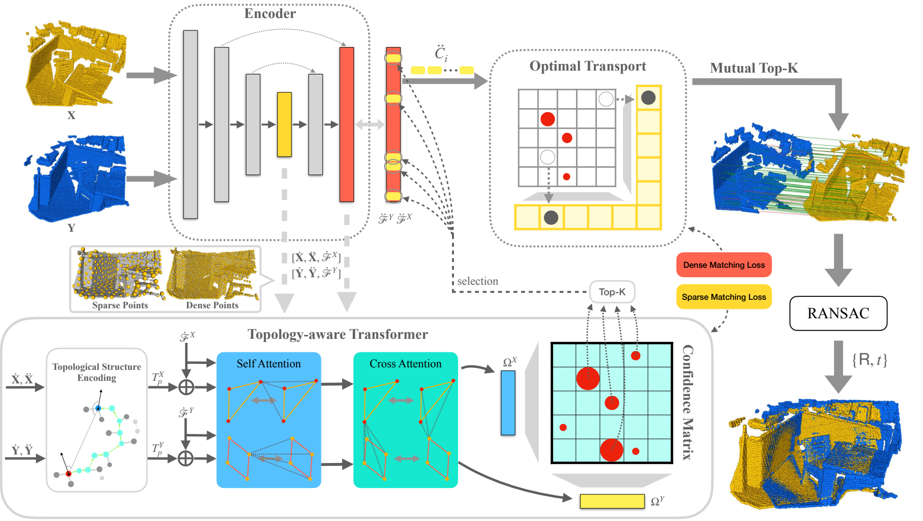

X
Y
Encoder
$\ddot{C}_i$
Optimal Transport
Mutual Top-K
$\dot{\mathscr{F}}^Y$ $\dot{\mathscr{F}}^X$
Sparse Points
Dense Points
$[\dot{\mathbf{X}}, \ddot{\mathbf{X}}, \dot{\mathscr{F}}^X]$
$[\dot{\mathbf{Y}}, \ddot{\mathbf{Y}}, \dot{\mathscr{F}}^Y]$
selection
Top-K
Dense Matching Loss
Sparse Matching Loss
RANSAC
$\{\mathrm{R}, t\}$
Topology-aware Transformer
$\dot{\mathbf{X}}, \ddot{\mathbf{X}}$
$\dot{\mathbf{Y}}, \ddot{\mathbf{Y}}$
Topological Structure Encoding
$T_p^X$
$T_p^Y$
$\dot{\mathscr{F}}^X$
$\dot{\mathscr{F}}^Y$
Self Attention
Cross Attention
$\Omega^X$
$\Omega^Y$
Confidence Matrix

**Fig. 3.** Illustration of our method. Dense and sparse points and their corresponding local features are extracted from the KPConv encoder. Points in different sparsities are then utilized to generate topological structure encoding which is merged with the sparse local features to perform self/cross-attention. We compute a confidence matrix from $\Psi^X$ and $\Psi^Y$ to obtain sparse point matches, which are used as constraints for selecting dense correspondence $\ddot{C}_i$. We employ a learnable optimal transport module for dense point refinement and further select the mutual top-k matches as the final correspondence.

### 3.1 Problem Definition

Given two point clouds $\mathbf{X} \in \mathbb{R}^{m\times3}$ and $\mathbf{Y} \in \mathbb{R}^{n\times3}$ which share sufficient overlap, our task is to establish a set of correspondences $\mathbb{C}^* = \{(\mathbb{C}_{xi}, \mathbb{C}_{yi}) | \mathbb{C}_{xi} \in \mathbb{R}^3, \mathbb{C}_{yi} \in \mathbb{R}^3, i = 1, \ldots, \mathbf{t}\}$. Then the rigid transformation $\mathbf{T}_X^Y = \{\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3\}$ that aligns $\mathbf{X}$ to $\mathbf{Y}$ can be optimized by solving the following equation:

$$\mathbf{R}, \mathbf{t} = \min_{\mathbf{R},\mathbf{t}} \sum_{(\mathbb{C}_{xi}, \mathbb{C}_{yi}) \in \mathbb{C}^*} \|\mathbf{R} \cdot \mathbb{C}_{xi} + \mathbf{t} - \mathbb{C}_{yi}\|_2^2 \qquad (1)$$

### 3.2 Local Feature Encoder

KPConv is a typical local feature extractor that is proven to be feasible and efficient for the point cloud registration task [3, 16]. We follow these works and employ such convolution strategies to design a network $\kappa(\cdot)$ which consists of a chain of ResNet-based [14] convolutional layers and multiple downsampling layers.

$$\{\dot{\mathbf{X}}, \dot{\mathcal{F}}^X, \ddot{\mathbf{X}}, \ddot{\mathcal{F}}^X\} = \kappa(X), \quad \{\dot{\mathbf{Y}}, \dot{\mathcal{F}}^Y, \ddot{\mathbf{Y}}, \ddot{\mathcal{F}}^Y\} = \kappa(Y), \tag{2}$$

where $\dot{\mathbf{X}} \in \mathbb{R}^{m'' \times 3}$ and $\dot{\mathbf{Y}} \in \mathbb{R}^{n'' \times 3}$ are the down-sampled points from the sparse layer. $\dot{\mathcal{F}}^X \in \mathbb{R}^{m'' \times d_f}$ and $\dot{\mathcal{F}}^Y \in \mathbb{R}^{n'' \times d_f}$ are the corresponding features. $d_f$ is the feature dimension. Similarly, $\ddot{\mathbf{X}} \in \mathbb{R}^{m' \times 3}$, $\ddot{\mathbf{Y}} \in \mathbb{R}^{n' \times 3}$, $\ddot{\mathcal{F}}^X \in \mathbb{R}^{m' \times d_f}$, and $\ddot{\mathcal{F}}^Y \in \mathbb{R}^{n' \times d_f}$ are the points and their corresponding features from the dense layer. Note that instead of using grid sampling to sub-sample the point cloud, we employ Furthest Point Sampling (FPS) to ensure $\dot{\mathbf{X}} \in \ddot{\mathbf{X}}$ and $\dot{\mathbf{Y}} \in \ddot{\mathbf{Y}}$, which will facilitate our proposed sparse-through-dense encoding. It will be elaborated in Section 3.3.

### 3.3 Topology-Aware Transformer

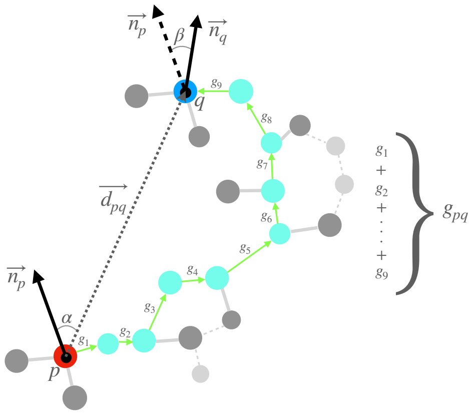


**Fig. 4.** An illustration of our topological structure encoding. Considering two sparse points $p$ and $q$ (represented by red and blue dots, respectively), we first build a KNN graph on dense points (grey dots), then the shortest path (i.e., geodesic distance, represented by green arrow) from $p$ to $q$ through dense points (aquamarine dots) can be computed progressively. The final topological structure encoding comprises the geodesic distance and the directional information.

Point-wise features extracted from the local feature encoder are merely dependent on the geometry relation in a local patch. Therefore, points at spatially disjoint positions that share similar local structures can be ambiguous in feature space due to the narrow perception field. To tackle this problem, Transformer has been widely used to strengthen global contextual information for the point cloud registration task [16, 27, 21, 35, 42, 41]. Early attempts directly feed the local geometry-based features to the Transformer, making the learned features spatially less distinctive at the global level [16, 35, 42]. This is later refined by either injecting point spatial information [41] or employing additional relative position encoding [27, 21] to the input features. Nevertheless, these methods only take into account the rough structural information of the sparse/super points, neglecting the underlying fine-grained topological structure.

Motivated by the analysis, we propose *TopFormer* which leverages the sparse-through-dense geodesic distance information, to allow the Transformer with sparse point-wise features input to perceive the dense-level topological information.

**Topological structure encoding.** To capture the fine-grained surface-based structural information in sparse points, we design a novel *topological structure encoding* in a sparse-through-dense manner. Its visualization is shown in Figure 4.

To encode the dense point topological structure into sparse features, we first construct a dense graph $\mathcal{G}_x$ on the dense points $\ddot{\mathbf{X}}$ by connecting $\forall p \in \ddot{\mathbf{X}}$ to their K-nearest-neighbours (KNN) with a radius constraint $\tau$:

$$\mathcal{G}_x = (\ddot{\mathbf{X}}, \mathcal{E}), \tag{3}$$

where $\mathcal{E} = \{(i,j) | i, j \in \ddot{\mathbf{X}}, j \in KNN(i), \|i - j\|_2 < \tau\}$. And $\| \cdot \|_2$ is the Euclidean norm. Given two sparse points $p, q \in \dot{\mathbf{X}}$, we can further compute its geodesic distance $g_{pq} \in \mathbb{R}^1$ from $p$ to $q$ through performing shortest path algorithms such as Dijkstra's [9], Floyd-Warshall [11] and Johnson's algorithms [19] on the dense graph $\mathcal{G}$. Similar to previous methods [15], Dijkstra's algorithm is selected in our experiments. We propose to use point normal to strengthen the directional information between $p$ and $q$. However, directly utilizing the normal can be problematic since the normal estimated by Principal Component Analysis (PCA) is usually unoriented. On the other hand, the point cloud in the registration task can be in arbitrary poses, making it difficult to reorient the normal consistently. To deal with the dilemma, we take an alternative strategy and use the sine value of the angles involving the normals, ensuring the invariance against unoriented normals (see Figure 2). The *topological structure encoding* $\hat{\mathbf{t}}_{pq} \in \mathbb{R}^{d_f}$ can be defined as follows:

$$\hat{\mathbf{t}}_{pq} = \mathcal{S}([g_{pq}; \|\vec{d_{pq}}\|_2; \sin(\angle(\vec{n_p}, \vec{d_{pq}})); \sin(\angle(\vec{n_p}, \vec{n_q}))]), \tag{4}$$

where $\mathcal{S}(\cdot) \in \mathbb{R}^{4 \times d_f}$ is the sinusoidal mapping function [33], $[\cdot;\cdot]$ is the concatenation operation, $\vec{d_{pq}}$ is the vector from $p$ to $q$. $\vec{n_p}$ and $\vec{n_q}$ are the sparse point normals. Normals are estimated at the dense point level by performing PCA on the local neighborhoods $\mathcal{E}$ that surround each point. $\angle(\vec{a}, \vec{b})$ is the angle between $\vec{a}$ and $\vec{b}$.

**Topology-aware self-attention.** The self-attention module enables each point in the point cloud to interact with every other point within the point cloud. By leveraging this mechanism, we are able to jointly learn the local-based geometry features associated with the intrinsic global topological correlation among the sparse points. Given a point $p \in \dot{\mathbf{X}}$ and its corresponding feature $f_p$, we first compute its *topological structure encoding* according to Eq. (4) with respect to all points in $\dot{\mathbf{X}}$, forming $T_p \in \mathbb{R}^{m'' \times d_f}$. Then the self-attention output $S_p^X$ with respect to $p$ can be formulated as follows:

$$S_p^X = \text{softmax}(\frac{f_p W^Q \cdot (\dot{\mathcal{F}}^X W^K + T_p W^E)^T}{\sqrt{d_f}}) \cdot \dot{\mathcal{F}}^X W^V, \tag{5}$$

where $W^Q$, $W^K$, $W^V$, $W^E \in \mathbb{R}^{d_f \times d_f}$ are learnable matrices of weights.

**Mutual information exchange.** We also adopt cross-attention to enable feature interaction between the two point clouds, allowing the pair to perceive each other. Given the features $S^X \in \mathbb{R}^{m'' \times d_f}$ and $S^Y \in \mathbb{R}^{n'' \times d_f}$ outputted by the self-attention module, the cross-attention can be formulated as:

$$C(K, Q, V) = \text{softmax}(\frac{QW^Q \cdot (KW^K)^T}{\sqrt{d_f}}) \cdot VW^V, \tag{6}$$

then the $K$ (key), $Q$ (query), and $V$ (value) for $X$ are $S^X$, $S^Y$ and, $S^Y$, respectively. And similarly, for $Y$ are $S^Y$, $S^X$, and $S^X$, respectively.

### 3.4 Sparse Point Matching

To bridge the features $\varPsi^X$ and $\varPsi^Y$ extracted from the Transformer with the correspondence proposals, we utilize the normalized features $\varOmega_X = norm(\varPsi^X)$ and $\varOmega_Y = norm(\varPsi^Y)$ to compute a Gaussian correlation matrix $S \in \mathbb{R}^{m'' \times n''}$:

$$S = \exp(2(\varOmega_X \varOmega_Y^T - 1)). \tag{7}$$

We further use a dual-normalization operation [29] to suppress the ambiguity caused by the geometrically less distinguishable patches (e.g., large flat surface in scene data):

$$\bar{s}_{i,j} = \frac{s_{i,j}}{\sum_{k=1}^{m''} s_{i,k}} \cdot \frac{s_{i,j}}{\sum_{k=1}^{n''} s_{k,j}}. \tag{8}$$

Finally, the top-k largest value entries of $\bar{S}$ are selected to be the matched sparse point pairs:

$$\dot{C} = \{(\dot{x}_i, \dot{y}_j) | (i, j) \in \text{topk}_{i,j}(\bar{S})\}. \tag{9}$$

### 3.5 Dense Points Refinement

The points matched in the sparse level are only able to provide coarse correspondences, meaning that given an optimal match point pair $(\dot{x}_i, \dot{y}_j) \in \dot{C}$, $\dot{y}_i$ is not guaranteed to be in the vicinity of $\dot{x}_i$. We use the dense points to further refine the coarse correspondences. Following [42, 20], given a point cloud $X$, its dense points $\ddot{X}$ are assigned to their nearest sparse points. Then a local patch $P_i^X \in \mathbb{R}^{l \times 3}$ with respect to a sparse point $\dot{x}_i \in \dot{X}$ can be defined as follows:

$$P_i^X = \{\ddot{x} \in \ddot{X} | i = \text{argmin}_j(\|\ddot{x} - \dot{x}\|_2), \dot{x} \in \dot{X}\} \tag{10}$$

Similarly, the local patch $P_j^Y \in \mathbb{R}^{k \times 3}$ with respect to a sparse point $\dot{y}_j \in \dot{Y}$ can be constructed in the same way. Then the corresponding patch features $\varTheta_i^X$ and $\varTheta_j^Y$ are obtained from dense points' features $\ddot{\mathbf{X}}$ and $\ddot{\mathbf{Y}}$, respectively.

Given a coarse correspondence $(\dot{x}_i, \dot{y}_j)$, we use an optimal transport layer [30] to sort out the correspondences from dense point patches $\varTheta_i^X \in \mathbb{R}^{l \times d_f}$ and $\varTheta_j^Y \in \mathbb{R}^{k \times d_f}$.

Concretely, we compute the correlation matrix $O_i \in \mathbb{R}^{(l+1)\times(k+1)}$ with additional row and column as slack entries:

$$O_i = \begin{bmatrix} \Theta_i^X(\Theta_j^Y)^T & \mathbf{z} \\ \mathbf{z}^T & z \end{bmatrix}, \tag{11}$$

and all slack entries are filled with a learnable parameter $z$. Then the Sinkhorn Algorithm [31] is applied to $O_i$, allowing points without the target match (e.g., points at non-overlapping area) to match their corresponding slack entries. The dense matching score matrix $D_i \in \mathbb{R}^{l\times k}$ can be recovered from $O_i$ by dropping all slack entries (i.e., the last row and column). We select the final dense point correspondences $\ddot{C}_i \in \mathbb{R}^{l'\times k'}(l' \leqslant l, k' \leqslant k)$ by picking up the mutual top-k entries on $D_i$:

$$\ddot{C}_i = \{P_i^X(k), P_j^Y(l) | (k,l) \in \text{mutual\_topk}_{k,l}(D_i)\}. \tag{12}$$

The final output correspondences can be formulated as $\mathbb{C}^* = \bigcup_i^{|\dot{C}|} \ddot{C}_i$, which is the union of Eq. (12) throughout all matched sparse point correspondences.

### 3.6 Loss Function

Our supervision loss function $\mathcal{L} = \mathcal{L}_s + \lambda\mathcal{L}_d$ comprises two parts: sparse matching loss $\mathcal{L}_s$ and dense matching loss $\mathcal{L}_d$.

**Sparse matching loss.** For the sparse point features, we attempt to minimize the $\mathcal{L}_2$ distance of the matched point features in a metric learning manner. Similar to [16, 27], we employ a weighted circle loss to minimize the feature distance of a given $\omega_i^X \in \Omega_X$ and its ground truth matches $\omega_i^Y \in G_{X_i}^Y$ (positive pair), and meanwhile maximize the feature distance of $\omega_i^X$ and $\omega_i^Y \in \hat{G}_{X_i}^Y = \Omega_Y \setminus G_{X_i}^Y$ (negative pair). The ground truth matches $G_{X_i}^Y$ is defined as the corresponding features of the sparse points $\dot{y}_j \in \dot{\mathbf{Y}}$ whose dense patch $P_j^Y$ shares over 10% overlap ratio with $P_i^X$. As for $X$, we select $\omega_i^X$ with both positive and negative pairs, forming $\Lambda_X^Y$. The loss with respect to $X$ can be formulated as:

$$\mathcal{L}_s^X = \frac{1}{|\Lambda_X^Y|} \sum_{\omega_i^X \in \Lambda_X^Y} log[1 + \sum_{\omega_j^Y \in G_{X_i}^Y} e^{\lambda_i^j \beta_p^{i,j} \sigma_p^{i,j}} \cdot \sum_{\omega_k^Y \in \hat{G}_{X_i}^Y} e^{\beta_n^{i,k} \sigma_n^{i,k}}], \tag{13}$$

the positive pair $\sigma_p^{i,j} = d_i^j - \Delta_p$ and the negative pair $\sigma_n^{i,k} = \Delta_n - d_i^k$, where $d_i^j = \|\omega_i^X - \omega_j^Y\|_2$, and the positive and negative margins are $\Delta_p = 0.1$ and $\Delta_n = 1.4$, respectively. And the weights are defined as $\beta_p^{i,j} = \gamma\sigma_p^{i,j}$ and $\beta_n^{i,k} = \gamma\sigma_n^{i,k}$ for positive and negative pairs, respectively. An additional overlap weight $\lambda_i^j = \sqrt{o_j^i}$ is used to emphasize the positive pairs that share a higher overlapping ratio. $o_j^i$ is the overlap ratio between $P_i^X$ and $P_j^Y$. The total sparse point loss is formulated as $\mathcal{L}_s = (\mathcal{L}_s^X + \mathcal{L}_s^Y)/2$, where $\mathcal{L}_s^Y$ can be calculated in a similar way to Eq. (13) with respect to $Y$.

**Dense matching loss.** For the dense point, given a pair of matched patches $P_k = \{P_i^X, P_j^Y\}$, the dense matching loss aims to maximize the matching score $D_i$ at the entries of matched points $M_i$. The matched points $M_i$ are selected with a matching threshold $\tau_d$. For the unmatched points $U_i^X \in P_i^X$ and $U_j^Y \in P_j^Y$, the loss tries to maximize the scores at the corresponding slack entries. The dense matching loss is thus formulated as follows:

$$\mathcal{L}_{d,k} = - \sum_{(x,y)\in M_i} \log D^i_{x,y} - \sum_{x\in U_i^X} \log D^i_{x,k+1} - \sum_{y\in U_j^Y} \log D^i_{l+1,y}, \tag{14}$$

where $D^i$ is the dense matching score matrix. The total dense matching loss is the average sum of $N_d$ matched patches: $\mathcal{L}_d = \frac{1}{N_d}\sum_k^{N_d} \mathcal{L}_{d,k}$.

# 4 Experiments

## 4.1 Implementation

Our method is implemented in PyTorch and we use Adam optimizer during the training stage with an initial learning rate of 1e-4 and a decay rate of 0.95 for every epoch and a weight decay of 1e-6. All the training is conducted on an Nvidia A100 GPU. For the 3DMatch dataset, we train the network for 20 epochs with a batch size of 1, which requires approximately 24 hours. For the KITTI dataset, the network is trained for 160 epochs with a batch size of 1, which requires approximately 12 hours.

## 4.2 Indoor Scene: 3DMatch

The 3DMatch dataset [43] is an indoor scene dataset for the point cloud registration task. The dataset contains 46, 8, and 8 scenes for training, validating, and testing, respectively. Following [16], we further split the test set into two evaluation sets based on their overlapping ratio, namely 3DMatch, and 3DLoMatch. Specifically, after removing the scene pairs with an extremely low overlapping ratio (i.e., less than 10%), we categorize those with an overlapping ratio of less than 30% as 3DLoMatch, while the rest were included in the 3DMatch evaluation set.

We follow [16] and compare experimental results with state-of-the-art methods using different correspondence numbers: PerfectMatch [13], FCGF [7], D3Feat [3], SpinNet [1], Predator [16], YOHO [34], CoFiNet [42], and GeoTransformer [27].

**Evaluation metrics.** Following [16], we report three metrics: (1) Inlier Ratio (IR), (2) Feature Matching Recall (FMR), and (3) Registration Recall (RR) to evaluate the performance of our method. The first two evaluate the quality of the predicted correspondence under the ground truth transformation. IR measures the fraction of the correspondence with a distance less than 0.1m compared with the ground-truth transformation and FMR estimates the fraction of the point cloud whose inlier ratio is over 5%. RR calculates the fraction of the point cloud pairs whose transformation RMSE is less than 0.2m, showing the quality of the final putative alignment.

**Correspondence results.** As shown in Table 1, despite that the feature matching recall is nearly saturated for the 3DMatch evaluation set, with CoFiNet [42] achieving over 98%, our method still outperforms all state-of-the-art methods consistently by $0.4 \sim 0.6$ percentage points. Similarly, our performance on the 3DLoMatch evaluation

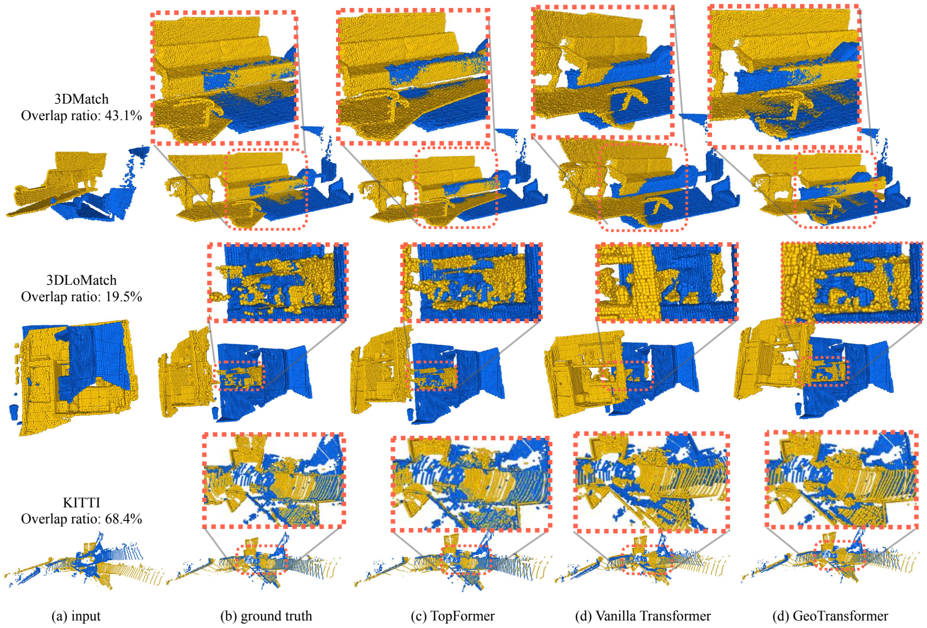


**Fig. 5.** Qualitative registration examples on both indoor and outdoor datasets with different overlap ratios.

set reaches around 88% consistently across most correspondence numbers. It surpasses GeoTransformer [27] by $0.2 \sim 1.4$ percentage points, despite the slightly weaker performance at the correspondence number of 250. For the inlier ratio, our method achieves comparable results at larger correspondence number cases (i.e., 5k and 2.5k). Although it fails to outperform GeoTransformer [27], it still dominates all other methods, surpassing Predator [16] by a large margin of 10.5 percentage points. Our method sees more significant improvement at low correspondence number cases (i.e., less than 2.5k). Despite the fluctuation at the 500 correspondence number, our other results outweigh those of GeoTransformer [27] with increasing $1.5 \sim 6.9$ percentage points. Moreover, our method surpasses the third-best methods (i.e., YOHO [34] or CoFiNet [42]) on both 3DMatch and 3DLoMatch by a large margin of $7.3 \sim 34.8$ percentage points.

**Registration results.** Registration recall directly reflects the success rate of the final registration, which is the dominant indicator of registration performance. We compare our registration results following [16]. Specifically, we run 50K RANSAC iterations on the established correspondence to estimate the final transformation. In general, Table 1 shows that our method outperforms state-of-the-art methods, regardless of the overlap ratio of the evaluation set. For the 3DMatch set, we obtained the best registration recall using 5k correspondence, achieving 92.4%. It surpasses GeoTransformer [27] (rank $2^{nd}$) and YOHO [34] (rank $3^{rd}$) by 0.4 and 1.6 percentage points, respectively. Improvements in the results with smaller correspondence numbers also demonstrate the effectiveness and robustness of our method. For the 3DLoMatch set, our method

**Table 1.** Evaluation results on 3DMatch and 3DLoMatch, with the top and second-ranking results highlighted in bold and underlined, respectively.

| | 3Dmatch | | | | | 3DLoMatch | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| #Samples | 5k | 2.5k | 1k | 500 | 250 | 5k | 2.5k | 1k | 500 | 250 |
| Feature Matching Recall % | | | | | | | | | | |
| PerfectMatch [13] | 95.0 | 94.3 | 92.9 | 90.1 | 82.9 | 63.6 | 61.7 | 53.6 | 45.2 | 34.2 |
| FCGF [7] | 97.4 | 97.3 | 97.0 | 96.7 | 96.6 | 76.6 | 75.4 | 74.2 | 71.7 | 67.3 |
| D3Feat [3] | 95.6 | 95.4 | 94.5 | 94.1 | 93.1 | 67.3 | 66.7 | 67.0 | 66.7 | 66.5 |
| SpinNet [1] | 97.6 | 97.2 | 96.8 | 95.5 | 94.3 | 75.3 | 74.9 | 72.5 | 70.0 | 63.6 |
| Predator [16] | 96.6 | 96.6 | 96.5 | 96.3 | 96.5 | 78.6 | 77.4 | 76.3 | 75.7 | 75.3 |
| YOHO [34] | 98.2 | 97.6 | 97.5 | 97.7 | 96.0 | 79.4 | 78.1 | 76.3 | 73.8 | 69.1 |
| CoFiNet [42] | 98.1 | 98.3 | 98.1 | 98.2 | 98.3 | 83.1 | 83.5 | 83.3 | 83.1 | 82.6 |
| GeoTransformer [27] | 97.9 | 97.9 | 97.9 | 97.9 | 97.6 | 87.2 | 87.2 | 86.8 | 87.4 | **87.1** |
| TopFormer | **98.6** | **98.7** | **98.7** | **98.4** | **98.5** | **87.8** | **88.1** | **88.2** | **87.6** | 86.8 |
| Inlier ratio% | | | | | | | | | | |
| PerfectMatch [13] | 36.0 | 32.5 | 26.4 | 21.5 | 16.4 | 11.4 | 10.1 | 8.0 | 6.4 | 4.8 |
| FCGF [7] | 56.8 | 54.1 | 48.7 | 42.5 | 34.1 | 21.4 | 20.0 | 17.2 | 14.8 | 11.6 |
| D3Feat [3] | 39.0 | 38.8 | 40.4 | 41.5 | 41.8 | 13.2 | 13.1 | 14.0 | 14.6 | 15.0 |
| SpinNet [1] | 47.5 | 44.7 | 39.4 | 33.9 | 27.6 | 20.5 | 19.0 | 16.3 | 13.8 | 11.1 |
| Predator [16] | 58.0 | 58.4 | 57.1 | 54.1 | 49.3 | 26.7 | 28.1 | 28.3 | 27.5 | 25.8 |
| YOHO [34] | 64.4 | 60.7 | 55.7 | 46.4 | 41.2 | 25.9 | 23.3 | 22.6 | 18.2 | 15.0 |
| CoFiNet [42] | 49.8 | 51.2 | 51.9 | 52.2 | 52.2 | 24.4 | 25.9 | 26.7 | 26.8 | 26.9 |
| GeoTransformer [27] | **71.9** | 75.2 | 76.0 | 82.2 | 85.1 | **43.5** | 45.3 | 46.2 | 52.9 | 57.7 |
| TopFormer | 71.7 | **76.6** | **82.8** | **82.3** | **87.0** | 37.2 | **45.4** | **53.1** | **56.7** | **59.2** |
| Registration Recall % | | | | | | | | | | |
| PerfectMatch [13] | 78.4 | 76.2 | 71.4 | 67.6 | 50.8 | 33.0 | 29.0 | 23.3 | 17.0 | 11.0 |
| FCGF [7] | 85.1 | 84.7 | 83.3 | 81.6 | 71.4 | 40.1 | 41.7 | 38.2 | 35.4 | 26.8 |
| D3Feat [3] | 81.6 | 84.5 | 83.4 | 82.4 | 77.9 | 37.2 | 42.7 | 46.9 | 43.8 | 39.1 |
| SpinNet [1] | 88.6 | 86.6 | 85.5 | 83.5 | 70.2 | 59.8 | 54.9 | 48.3 | 39.8 | 26.8 |
| Predator [16] | 89.0 | 89.9 | 90.6 | 88.5 | 86.6 | 59.8 | 61.2 | 62.4 | 60.8 | 58.1 |
| YOHO [34] | 90.8 | 90.3 | 89.1 | 88.6 | 84.5 | 65.2 | 65.5 | 63.2 | 56.5 | 48.0 |
| CoFiNet [42] | 89.3 | 88.9 | 88.4 | 87.4 | 87.0 | 67.5 | 66.2 | 64.2 | 63.1 | 61.0 |
| GeoTransformer [27] | 92.0 | 91.8 | 91.8 | **91.4** | 91.2 | 75.0 | 74.8 | 74.2 | 74.1 | 73.5 |
| TopFormer | **92.4** | **92.1** | **92.1** | **91.4** | **91.4** | **75.1** | **75.7** | **74.3** | **74.3** | **74.0** |

exceeds GeoTransformer [27] (rank $2^{nd}$) by $0.1 \sim 0.9$ percentage points and CoFiNet [42] (rank $3^{rd}$) by at least 7.6 percentage points.

### 4.3 Outdoor Scene Data: KITTI

The KITTI odometry dataset [12] is an outdoor driving scene dataset consisting of 11 LiDAR-scanned sequences. Following [16, 7], we split the sequences into three parts: 0-5, 6-7, and 8-10 for training, validating, and testing, respectively. According to [7, 3], the provided ground truth poses are refined with ICP [4] and we only evaluate those point cloud pairs that are less than 10 meters away from each other.

**Evaluation metrics.** We follow [16] and report the Relative Translation Error (RTE), Relative Rotation Error (RRE), and Registration Recall (RR). RTE is defined as the $\mathcal{L}_2$ norm of the deviation between the predicted and ground truth translation vector, i.e., $RTE = \|t - t_{gt}\|_2$. The RRE is defined as the follows:

$$RRE = \arccos(\frac{tr(R^T \cdot R_{gt} - 1)}{2}), \tag{15}$$

where $tr(\cdot)$ is the trace operator. $R$ and $R_{gt}$ are the predicted rotation matrix and the ground truth rotation matrix, respectively. RR is calculated as the fraction of the point cloud pairs whose RRE and RTE are both below certain thresholds (i.e., $RRE < 5^\circ$ and $RTE < 2m$).

**Table 2.** Evaluation results on KITTI, with the top and second-ranking results highlighted in bold and underlined, respectively.

| Model | RTE (cm) | RRE (°) | RR (%) |
|---|---|---|---|
| 3DFeat-Net[39] | 25.9 | **0.25** | 96.0 |
| FCGF[7] | 9.5 | 0.30 | 96.6 |
| D3Feat[3] | 7.2 | 0.30 | **99.8** |
| SpinNet[1] | 9.9 | 0.47 | 99.1 |
| Predator[16] | 6.8 | 0.27 | **99.8** |
| CoFiNet[42] | 8.2 | 0.41 | **99.8** |
| GeoTransformer[27] | 7.4 | 0.27 | **99.8** |
| TopFormer | **6.7** | 0.27 | **99.8** |

**Registration results.** We compare our results with the state-of-the-art methods: 3DFeat-Net[39], FCGF [7], D3Feat [3], SpinNet [1], CoFiNet [42], and GeoTransformer [27] in Table 2. For registration recall, all state-of-the-art approaches perform strongly on the dataset and have achieved over 95%. A majority of them even achieved a nearly saturated score of 99.8%. Interestingly, our method also shows strong capabilities in outdoor scenarios, reaching 99.8%. For RRE, TopFormer is slightly weaker (0.02 degree) than 3DFeat-Net [3]. However, it surpasses D3Feat-Net [3] in terms of RTE by a large margin of 19.2 cm. Moreover, the proposed TopFormer also beats the strong competitor Predator [16] by 0.1 cm, thus highlighting the robustness of our proposed approach.

### 4.4 Ablation Study

To demonstrate how each component contributes to the overall performance, we conduct a series of ablation experiments and follow [27] to report Inlier Ratio (IR), Feature Matching Recall (FMR), Registration Recall (RR), and an additional metric Pair Inlier Ratio (PIR) that measures the fraction of the sparse point patches with the actual overlap. The ablation experiments are trained and evaluated on 3DMatch with both high and low overlap benchmarks. The results are listed in Table 3 which can answer the following questions:

**Is the *Topological Structure Encoding* useful?** Removing the proposed *Topological Structure Encoding* from TopFormer makes it degenerate into a Vanilla Transformer. It fails to take into account either the global position information or the structural information of the input feature, causing ambiguity when dealing with locally similar but globally faraway points. The evaluation results in Table 3 show that the performance of the Vanilla Transformer is significantly weaker than the proposed TopFormer. Specifically, the registration recall of the Vanilla Transformer is inferior to that of TopFormer

**Table 3.** Ablation study results on 3DMatch & 3DLoMatch, with the top result highlighted in bold.

| | 3Dmatch | | | | 3DLoMatch | | | |
|---|---|---|---|---|---|---|---|---|
| Method | PIR | FMR | IR | RR | PIR | FMR | IR | RR |
| Vanilla Tranformer | 79.5 | 97.5 | 63.1 | 89.7 | 45.0 | 85.2 | 31.6 | 67.4 |
| w/o normal | 84.4 | 98.5 | 67.9 | 91.9 | 48.4 | 87.9 | 35.1 | 73.9 |
| w/o geodesic distance | 83.6 | 98.1 | 69.9 | 90.3 | 48.8 | 87.5 | 37.1 | 72.3 |
| Default | **85.3** | **98.6** | **71.7** | **92.4** | **51.8** | **87.8** | **37.2** | **75.1** |

by 2.7 percentage points and 7.7 percentage points for 3DMatch and 3DLoMatch, respectively. All other metrics also show the remarkable capability of the proposed *Topological Structure Encoding*. In particular, benefiting from the strong topological information captured by this module, the inlier ratio of Vanilla Transformer is surpassed by TopFormer by a large margin of 8.6 percentage points and 5.6 percentage points for 3DMatch and 3DLoMatch, respectively. Visual comparisons between the Vanilla Transformer and TopFormer are shown in Figure 5.

**Does the sine-based directional information make sense?** The directional relationship information between two sparse points is depicted by the sine value of the angle involving the normals of the corresponding sparse points. Here we try to remove both $\cos(\angle(\vec{n_p}, \vec{d_{pq}}))$ and $\cos(\angle(\vec{n_p}, \vec{n_q})$ from the *Topological Structure Encoding* while remaining the other components. Results show that the performance is weakened for both high and low overlap benchmarks by 0.5 percentage points and 1.2 percentage points, respectively. This confirms the usefulness of our introduced sine-based directional information between two sparse points.

**How important is the *geodesic distance*?** The topological information of the point cloud is mainly captured by the geodesic distance. To demonstrate the effectiveness of the injected surface-based structural information, the geodesic distance is removed from the *Topological Structure Encoding* while maintaining the other components. From Table 3, the registration recall observes a significant drop from 92.4% to 90.3% and from 75.1% to 72.3% for 3DMatch and 3DLoMatch, respectively. These results demonstrate the advantage of our introduced topological information during the registration task.

## 5 Conclusion

We have presented TopFormer, a novel point cloud registration approach that leverages surface-based structural information to learn robust topology-aware representations for feature matching. We designed a topology structure encoding to capture the point-pair surface-based structure in a sparse-through-dense manner, enabling the learned sparse feature descriptors to take into account the dense point surface-based structure. We evaluate our method on both indoor and outdoor datasets. Experimental results show our approach outperforms the state-of-the-art methods. We also conduct ablation studies to analyze and verify the effectiveness of the key components in TopFormer.

## References

1. Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y.: Spinnet: Learning a general surface descriptor for 3d point cloud registration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11753–11762 (2021)
2. Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: Robust & efficient point cloud registration using pointnet. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7163–7172 (2019)
3. Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C.L.: D3feat: Joint learning of dense detection and description of 3d local features. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6359–6367 (2020)
4. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures. vol. 1611, pp. 586–606. Spie (1992)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 213–229. Springer (2020)
6. Chen, C.F.R., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 357–366 (2021)
7. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8958–8966 (2019)
8. Deng, H., Birdal, T., Ilic, S.: Ppfnet: Global context aware local features for robust 3d point matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 195–205 (2018)
9. Dijkstra, E.W.: A note on two problems in connexion with graphs:(numerische mathematik, 1 (1959), p 269-271) (1959)
10. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 998–1005. Ieee (2010)
11. Floyd, R.W.: Algorithm 97: shortest path. Communications of the ACM **5**(6), 345 (1962)
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)
13. Gojcic, Z., Zhou, C., Wegner, J.D., Wieser, A.: The perfect match: 3d point cloud matching with smoothed densities. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5545–5554 (2019)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. He, T., Huang, H., Yi, L., Zhou, Y., Wu, C., Wang, J., Soatto, S.: Geonet: Deep geodesic networks for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6888–6897 (2019)
16. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: Predator: Registration of 3d point clouds with low overlap. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 4267–4276 (2021)
17. Huang, X., Mei, G., Zhang, J.: Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11366–11374 (2020)

18. Jiang, J., Lu, X., Zhao, L., Dazaley, R., Wang, M.: Masked autoencoders in 3d point cloud representation learning. IEEE Transactions on Multimedia (2023)
19. Johnson, D.B.: Efficient algorithms for shortest paths in sparse networks. Journal of the ACM (JACM) **24**(1), 1–13 (1977)
20. Li, J., Chen, B.M., Lee, G.H.: So-net: Self-organizing network for point cloud analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9397–9406 (2018)
21. Li, Y., Harada, T.: Lepard: Learning partial point cloud matching in rigid and deformable scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5554–5564 (2022)
22. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
23. Lu, X., Chen, H., Yeung, S.K., Deng, Z., Chen, W.: Unsupervised articulated skeleton extraction from point set sequences captured by a single depth camera. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
24. Lu, X., Deng, Z., Luo, J., Chen, W., Yeung, S.K., He, Y.: 3d articulated skeleton extraction using a single consumer-grade depth camera. Computer Vision and Image Understanding **188**, 102792 (2019)
25. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2906–2917 (2021)
26. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
27. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K.: Geometric transformer for fast and robust point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11143–11152 (2022)
28. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
29. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. Advances in neural information processing systems **31** (2018)
30. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
31. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics **21**(2), 343–348 (1967)
32. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6411–6420 (2019)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
34. Wang, H., Liu, Y., Dong, Z., Wang, W.: You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 1630–1641 (2022)
35. Wang, Y., Solomon, J.M.: Deep closest point: Learning representations for point cloud registration. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3523–3532 (2019)
36. Wang, Y., Solomon, J.M.: Prnet: Self-supervised learning for partial-to-partial registration. Advances in neural information processing systems **32** (2019)

37. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
38. Xu, H., Liu, S., Wang, G., Liu, G., Zeng, B.: Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3132–3141 (2021)
39. Yew, Z.J., Lee, G.H.: 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In: Proceedings of the European conference on computer vision (ECCV). pp. 607–623 (2018)
40. Yew, Z.J., Lee, G.H.: Rpm-net: Robust point matching using learned features. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11824–11833 (2020)
41. Yew, Z.J., Lee, G.H.: Regtr: End-to-end point cloud correspondences with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6677–6686 (2022)
42. Yu, H., Li, F., Saleh, M., Busam, B., Ilic, S.: Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. Advances in Neural Information Processing Systems **34**, 23872–23884 (2021)
43. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1802–1811 (2017)
44. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16259–16268 (2021)