scientific reports

Check for updates

OPEN

HarSoNet: a two-stage point cloud registration method integrating soft and hard matching

Qiongdan Huang[1]✉, Jiapeng Wang[1], Jiejing Han[1,2] & Shilin Kang[1,2]

The purpose of point cloud registration is to determine the transformation parameters among multiple partially overlapping point clouds, and it plays an important role in various scenarios such as simultaneous localization and mapping (SLAM), scene reconstruction, industrial manufacturing and so on. However, due to the unordered and irregular nature of point clouds, accurately establishing correspondences poses a significant challenge. Coarse-to-fine methods, consisting of coarse and fine matching stages, have become popular in point cloud registration due to their effectiveness in handling repeatable keypoints. However, these methods are highly sensitive to the correspondences generated during the coarse-matching stage, where low-quality correspondences can lead to complete registration failure. Furthermore, the hard matching approach employed in coarse and fine matching stages often introduces a large number of outliers into the established correspondences. To overcome these challenges, this study introduces HarSoNet, a two-stage Hard-to-Soft Network designed for end-to-end point cloud registration. In the hard matching stage, the model incorporates a hybrid similarity fusion module, which combines similarity scores obtained from different algorithms to establish superpoint correspondences. These superpoint correspondences, along with their neighboring points, are then grouped into fuzzy patch correspondences. In the soft matching stage, patch correspondences are refined into point correspondences by calculating and adjusting the similarity matrix for each fuzzy patch. Finally, all local point correspondences are aggregated into global correspondences, and the transformation parameters are computed using the weighted singular value decomposition (SVD) algorithm. Experimental results demonstrate that HarSoNet achieves Error(R) = 1.376 and Error(t) = 0.015 on noisy, partially overlapping point clouds, demonstrating high registration accuracy and strong generalization performance.

Keywords Point cloud registration, Coarse-to-fine correspondences, Soft matching, Hard matching

Point cloud registration enables the alignment of point clouds from different data sources or acquisition perspectives, providing critical data support for decision-making, design, and research in fields such as industrial manufacturing, autonomous driving, and robotic navigation[1,2]. In particular, laser manufacturing systems require multiple scans of large and complex components from various directions, followed by precise alignment of these point clouds to construct a 3D surface model of the components[3,4]. Nevertheless, point cloud registration presents a multitude of challenges and difficulties. On one hand, the sparsity and noise interference in point cloud data make it challenging to accurately identify corresponding points. On the other hand, existing methods often exhibit low efficiency and struggle to ensure accuracy when processing complex scenes and large-scale point cloud data.

Traditional point cloud registration methods, such as ICP[5], NDT[6], and FGR[7], perform registration by iteratively optimizing an error metric. These methods are simple to implement and widely applicable but may suffer from slow convergence and a tendency to fall into local minima under poor initial registration or partial overlap. In recent years, numerous learning-based point cloud registration methods[8–16] have utilized rotation-invariant features to establish correspondences between two point clouds, eliminate outlier matches, and apply robust estimator to compute the optimal transformation parameters, thus achieving accurate point cloud registration. It is noteworthy that establishing accurate correspondences constitutes the critical determinant of registration success, whose quality fundamentally governs both the stability of subsequent transformation estimation and the ultimate registration precision.

[1]School of Communication and Information Engineering, Xi'an University of Post and Telecommunications, 710121 Xi'an, China. [2]Jiejing Han and Shilin Kang contributed equally. ✉email: limitless010@163.com

Soft matching correspondence-based approaches permit each point to establish multiple potential correspondences, each associated with a probability or weighting factor. DCP[10]determines soft correspondences by calculating similarity scores between point-wise features of two point clouds. Since not every point is repeatable, RPM-NET[12]assigns weights to each soft matching correspondence to select the final match. ROP-NET[13]refines the similarity matrix iteratively to achieve high-quality soft matching correspondences. However, these methods perform poorly in large-scale real-world tests and fail to resolve the low-overlap registration problem. Hard matching methods require a unique match for each point in the source point cloud. D3 Feat[11]employs a density-invariant keypoint selection strategy to identify repeatable keypoints between two point clouds for registration. Predator[14] achieves robust registration results for low-overlap problems by predicting the overlap region of two point clouds and selecting correspondences with high matching scores within that region. However, these approaches face challenges in identifying accurate correspondences when encountering substantial deformations or notable local geometric variations.

Inspired by the coarse-to-fine strategy in image matching[17–19], CoFiNet[15]establishes correspondences between two point clouds using a two-stage matching process. In the coarse matching stage, the input point cloud is downsampled to generate superpoints by introducing sparsity. The similarity matrix, constructed from the superpoint features, is iteratively refined using Sinkhorn to determine hard matching correspondences. In the fine matching stage, superpoint correspondences are combined with neighboring points to form fuzzy patch correspondences. Finally, meaningless entries in the similarity matrix, constructed from block correspondences, are masked by the density-adaptive matching module to obtain final point correspondences. This coarse-to-fine matching mechanism mitigates the risk of losing correspondences due to downsampling, enhancing the efficiency and robustness of the identified correspondences. However, this approach relies heavily on the accuracy of the superpoint correspondences obtained during the coarse-matching stage. GeoTrans[16] employs a Transformer model to learn rotation-invariant geometric features for superpoint matching, improving the quality of superpoint correspondences. However, as both matching stages are based on hard matching, other potential relationships between points are overlooked, leading to final correspondences that are still not sufficiently accurate or robust.

This paper addresses the shortcomings of existing two-stage matching methods, such as the presence of numerous outliers, by proposing a network model that combines soft and hard matching. In the hard matching stage, a hybrid similarity fusion module is employed to assess feature similarity by combining scaled dot product and pairwise distance similarity results, thereby obtaining superpoint correspondences. In the soft matching stage, each point in the two slices of the point cloud is assigned to the nearest superpoint in geometric space to form fuzzy patch correspondences. As shown in Fig. 1, points of the same color in the two point clouds form the patch correspondences. By restricting the estimated soft assignments between the two point clouds to the range of patch correspondences, the probability of false matches is effectively reduced. The final point correspondences are then obtained by calculating and adjusting the similarity matrix for each patch. Finally, all local point correspondences are combined into global point correspondences, and the final transformation parameters are computed using weighted SVD.

The main contributions of this paper are as follows:

1) A combined soft and hard matching approach is proposed, where high-quality superpoint correspondences from the hard matching stage provide a clear search direction for soft matching, which assigns multiple potential matching probabilities to each point along that direction. The robust point correspondences established by the combination of two-stage soft and hard matching lead to fast and reliable matching.
2) A hybrid similarity fusion module is designed, where distance-based similarity captures the spatial distribution and local similarity of features, while matrix multiplication-based similarity captures the global structure of the point cloud and feature interactions. The combination of both methods allows for a more comprehensive evaluation of feature similarity.

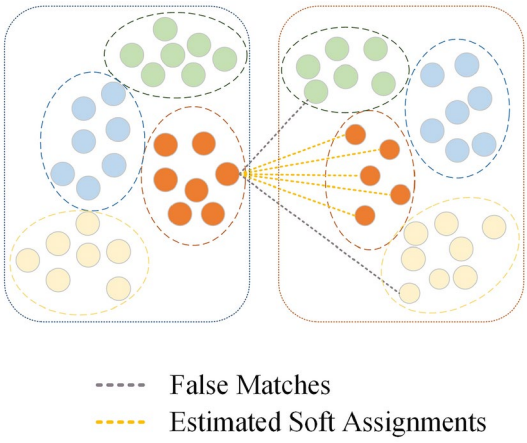

Fig. 1. Patch correspondences in the soft matching stage.

Related work

Most point cloud registration algorithms can be broadly divided into four key steps: feature extraction, correspondence search, outlier filtering, and transformation parameter estimation.

Feature extraction methods

Many algorithms utilize unique feature descriptors to determine point cloud correspondences, resulting in more accurate registration results[16,20–25]. SpinNet[21]voxelizes point clouds into spherical representations and leverages meticulously designed cylindrical convolutions to extract rich local features, demonstrating strong generalization capability in unseen scenarios. However, its voxel representation incurs significant computational overhead. GeoTrans[16]encodes rotation-invariant features, such as angles and distances between superpoints, and improves feature representation by capturing global relationships via the Transformer. Nevertheless, these methods fail to account for the fact that rotationally isovariant features retain the orientation information of the point cloud. YOHO[23]and RoReg[25]simultaneously encode both rotationally invariant and isovariant features, enabling fast and robust registration. Recent research on adversarial attacks in 3D point clouds[26] has shown that manifold-constrained methods can enhance robustness to local geometric deformations by limiting perturbations in the feature space.

Correspondence search methods

In the presence of noise interference and partial overlap, constructing robust point cloud correspondences poses significant challenges. Reference[27]introduces the concept of hardening soft-assignment, dynamically adjusting the strictness of matching through Gaussian kernels and an annealing strategy. This approach encourages the model to prioritize high-confidence correspondences during inference, effectively suppressing noise and outliers. MCSVR[28], building on the coarse-to-fine framework of CoFiNet[15], innovatively employs a region-level Gaussian mixture model to represent the geometric distribution of point clouds. This method quickly filters potential matching regions and establishes precise hard correspondences within local areas by balancing local geometric details and global distribution characteristics. Unlike these existing approaches, the proposed algorithm first establishes hard correspondences between superpoints, forming region-level correspondences through Euclidean neighborhood aggregation. Then, within a constrained search space, soft assignment is applied to allocate multiple matching weights to each point based on feature similarity. This "hard-matching-guided, soft-assignment-refined" mechanism maintains the determinacy of hard matching while leveraging the fault tolerance of soft assignment, thereby enhancing the robustness of point cloud registration.

Outlier filtering methods

Existing learning-based methods[29–31]first filter initial correspondences using deep feature similarity and then rely on RANSAC or regression networks to estimate transformation parameters. However, these approaches neither fully exploit spatial information in geometric transformations nor consider the overall spatial relationships among inliers. Spatial compatibility-based methods[32–34]incorporate spatial consistency constraints during feature extraction and outlier filtering. Nevertheless, they still depend on coarse correspondences as input and cannot achieve end-to-end point cloud registration. Notably, SymAttack[35] introduces a novel symmetry-aware attack framework that generates perturbations while preserving the symmetry of the point cloud. The outliers produced by such perturbations are highly inconspicuous-not only are they subtle, but they also maintain the original shape's symmetry. This characteristic makes traditional geometry-based outlier filtering methods ineffective in detecting these perturbations, posing new challenges to the robustness of the registration process.

Transformation parameter estimation methods

The coupling between the rotation matrix and translation vector complicates the estimation of rigid transformation parameters. This occurs because variations in one parameter (either in the rotation matrix or translation vectors) may influence the estimation of the other, potentially causing bias or errors in the final registration result. To address the problem of parameter interference, DetarNet[36]separates the estimation of rotation and translation into two stages. Once the interference from translation vectors is removed, the rotation parameters are determined using singular value decomposition (SVD). FINet[37] employs a two-branch structure to encourage the model to separately extract features for the two parameters and then applies a regression network to predict the final transformation. However, these methods often fail to fully exploit the local geometric structure of the point cloud, leading to a significant number of outliers in the established correspondences.

Methodology

Problem description

Let the source point cloud be $P = \{p_1, p_2, \ldots, p_n\}$ and the reference point cloud be $Q = \{q_1, q_2, \ldots, q_m\}$ representing the point sets of two point clouds to be aligned, where n and m represent the number of points in P and Q, respectively. The goal of the point cloud registration is to find an optimal rigid transformation $\mathrm{T} = (\mathrm{R}, \mathrm{t})$ that best aligns the source point cloud P with the reference point cloud Q, where $\mathrm{R} \in SO(3)$ is the rotation matrix (the special orthogonal group) and $\mathrm{t} \in \mathbb{R}^3$ is the translation vector. The optimal rigid transformation T is obtained by minimizing the distance between the real corresponding points (p_i^*, q_j^*):

$$\min_{\mathrm{R},\mathrm{t}} \sum_{\left(p_i^*, q_j^*\right) \in C^*} ||\mathrm{R} \cdot p_i^* + \mathrm{t} - q_j^*||_2^2 \tag{1}$$

C^* is the set of real correspondences between the two point clouds. The method proposed in this paper leverages the speed of hard matching and the robustness of soft matching by combining two-stage soft and hard matching, ensuring the entire matching process produces robust correspondences while maintaining efficiency.

Network architecture

The overall architecture of the proposed method is depicted in Fig. 2. The backbone network performs a downsampling of the input point clouds P and Q while simultaneously extracting their features. The hard matching stage establishes hard correspondences between the superpoints by computing the similarity of superpoint features using the hybrid similarity fusion module. The superpoint correspondences are extended to patch correspondences by combining their neighboring points. The soft matching stage refines these patch correspondences into point correspondences by computing and adjusting the similarity matrix of the patches. Finally, all local point correspondences are pooled into global correspondences, and the final transformation parameters are computed using the weighted SVD algorithm.

Point cloud preprocessing

Point clouds typically contain a large number of points, and high point cloud density can lead to significant computational costs when calculating distances or performing feature matching. Reducing the number of points through downsampling effectively reduces computational complexity, thereby improving registration efficiency. In this paper, we use the KPConv-FPN backbone network[38,39] to downsample the original point clouds P and Q, extracting their corresponding features. Through the first level of downsampling, we derive the dense point sets $\tilde{P}$ and $\tilde{Q}$, along with their feature representations $\tilde{F}^P \in \mathbb{R}^{|\tilde{P}|\times d}$ and $\tilde{F}^Q \in \mathbb{R}^{|\tilde{Q}|\times d}$. At the final level of downsampling, we obtain the superpoint sets $\hat{P}$ and $\hat{Q}$, as well as their feature representations $\hat{F}^P \in \mathbb{R}^{|\hat{P}|\times \hat{d}}$ and $\hat{F}^Q \in \mathbb{R}^{|\hat{Q}|\times \hat{d}}$. Here, $|\tilde{P}|$ and $|\tilde{Q}|$ denote the number of dense points, while $\tilde{d}$ represents the dimensionality of the dense point features. Similarly, $|\hat{P}|$ and $|\hat{Q}|$ indicate the number of superpoints, with $\hat{d}$ denoting the dimensionality of the superpoint features. Each point $\tilde{p}$ in the dense point $\tilde{P}$ is mapped to the nearest superpoint in geometric space, forming a local patch M_i^P:

$$M_i^P = \left\{\tilde{p} \in \tilde{P} \,\middle|\, i = \arg\min_j \left(\|\tilde{p} - \hat{p}_j\|_2\right), \hat{p}_j \in \hat{P}\right\} \tag{2}$$

The feature F_i^P of patch M_i^P is composed of the features $\tilde{F}^{\tilde{p}}$ of the points $\tilde{p}$ within M_i^P :

$$F_i^P = \left\{\tilde{F}^{\tilde{p}} \in \tilde{F}^P \,\middle|\, i = \arg\min_j \left(\|\tilde{p} - \hat{p}_j\|_2\right), \hat{p}_j \in \hat{P}\right\} \tag{3}$$

Similarly, the local patch M_i^Q of point cloud Q and its corresponding feature F_i^Q are computed using Eq. (2) and Eq. (3).

Hard matching stage

In this paper, we establish hard correspondences between superpoints by encoding their *rotation invariant embedding* and applying the proposed *hybrid similarity fusion module*. The framework of the hard matching stage is illustrated in Fig. 3. First, rotation invariant embedding are encoded for superpoints $\hat{P}$ and $\hat{Q}$, respectively. Next, these features, along with those extracted by the backbone network $\hat{F}^P$ and $\hat{F}^Q$, are fed into the Transformer. This process, comprising *rotation invariant embedding* and Transformer, is defined as the *Rotation Invariant Transformer Embedding*. Finally, hard correspondences between superpoints are obtained using the *hybrid similarity fusion module*.

Rotation invariant transformer embedding

Most previous directly input the features extracted by neural networks into the Transformer, where some information may be redundant or unnecessary, reducing the geometric distinctiveness of the model. Studies[13,16,40] have shown that incorporating geometric features, such as point cloud rotational invariance, into the Transformer enhances its focus on critical alignment features, thereby reducing ambiguities, mitigating outlier matches, and improving registration efficiency and performance.

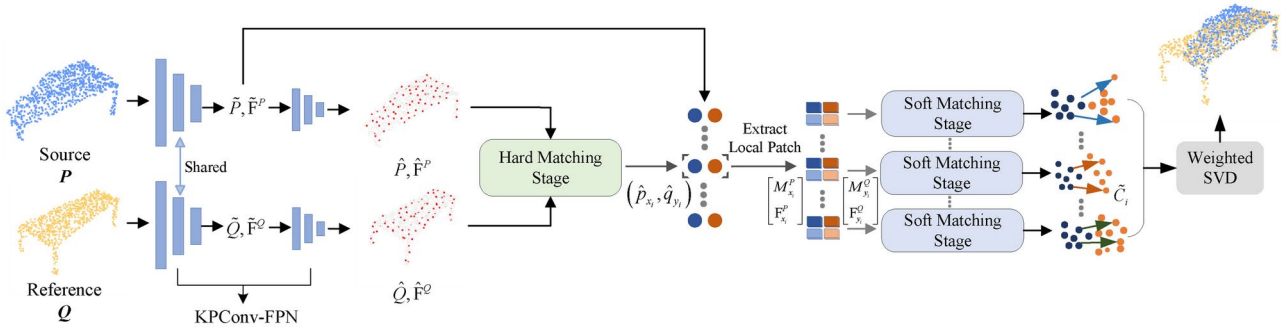

Fig. 2. HarSoNet network architecture.

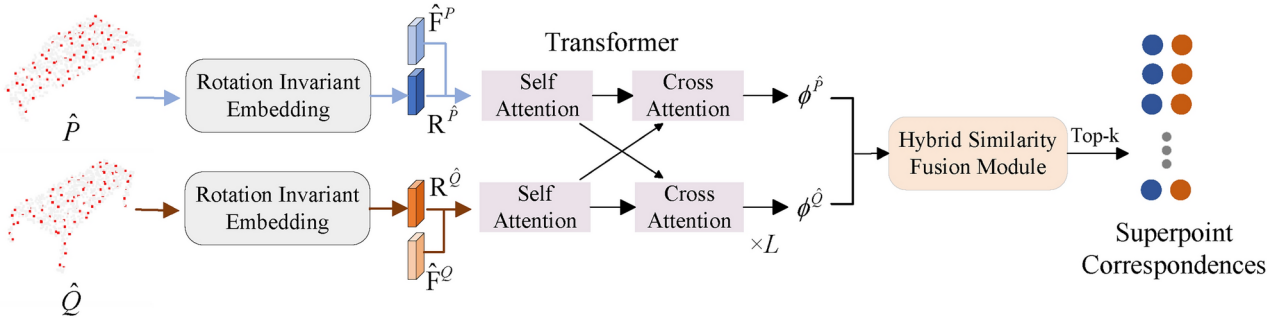

Fig. 3. Hard matching stage.

In this paper, we apply pair-wise distance embedding and triplet-wise angular embedding[16] for superpoints. The core idea is to extract rotation-invariant embedding (angles and distances), which remain constant under rigid transformations. Theoretically, these features ensure a stable representation of the same spatial entity's geometric properties across different viewpoints.

Given two superpoints $\hat{p}_i, \hat{p}_j \in \hat{P}$, the pair-wise distance embedding $r^D_{i,j} \in \mathbb{R}^f$ between them, with even-dimensional feature $r^D_{i,j,2r}$ and odd-dimensional feature $r^D_{i,j,2r+1}$, can be expressed as:

$$\begin{cases} r^D_{i,j,2r} = \sin\left(\frac{d_{i,j}}{10^{4\cdot 2r/f}}\right) \\ r^D_{i,j,2r+1} = \cos\left(\frac{d_{i,j}}{10^{4\cdot 2r/f}}\right) \end{cases} \tag{4}$$

where r denotes the index of the current dimension, f is the dimension of the feature, $d_{i,j} = ||\hat{p}_i - \hat{p}_j||_2$ is the Euclidean distance between the superpoints $\hat{p}_i$ and $\hat{p}_j$. σ_d is a hyperparameter that adjusts distance scaling. Eq. (4) represents the spatial distribution pattern of superpoints. As the dimension r increases, different frequency encodings are generated using sine and cosine functions. High-frequency components capture subtle distance variations, while low-frequency components encode large-scale distance patterns, enabling adaptive perception of multi-scale geometric structures.

Assuming $\hat{p}_c (1 \le c \le k)$ represents the k nearest points in the neighbourhood of superpoint $\hat{p}_i$, we define $\Delta_{i,j} = \hat{p}_i - \hat{p}_j$. For each $\hat{p}_c$, $\alpha^A_{i,j,c} = \angle(\Delta_{c,j}, \Delta_{j,i})$, which denotes the angle between vector $\Delta_{c,j}$ and $\Delta_{j,i}$. The triplet-wise angular embedding $r^A_{i,j,c}$ between superpoint $\hat{p}_i$, its neighbourhood point $\hat{p}_c$, and superpoint $\hat{p}_j$ can be represented by the even-dimensional feature $r^A_{i,j,c,2l}$ and the odd-dimensional feature $r^A_{i,j,c,2l+1}$:

$$\begin{cases} r^A_{i,j,c,2l} = \sin\left(\frac{\alpha^A_{i,j,c}}{10^{4\cdot 2l/f}}\right) \\ r^A_{i,j,c,2l+1} = \cos\left(\frac{\alpha^A_{i,j,c}}{10^{4\cdot 2l/f}}\right) \end{cases} \tag{5}$$

l denotes the index of the current dimension, and σ_d is a hyperparameter that adjusts for angular changes. Eq. (5) captures the local geometric morphology of superpoints. Similar to pair-wise distance embedding, it employs sine and cosine functions to generate multi-frequency scale coverage as the dimension l increases. High-frequency components distinguish subtle angular variations, while low-frequency components encode macroscopic geometric patterns. Compared to traditional linear projection methods, the nonlinear mapping of sine and cosine functions better adapts to complex geometric structures.

Finally, the *rotation invariant embedding* between the superpoint $\hat{p}_i$ and $\hat{p}_j$ can be expressed as:

$$r_{i,j} = r^D_{i,j}\mathrm{W}^D + \max_c \left\{ r^A_{i,j,c}\mathrm{W}^A \right\} \tag{6}$$

Here, $r_{i,j} \in R^{\hat{P}}$, W^D and $\mathrm{W}^A \in \mathbb{R}^{f\times f}$, are projection matrices for distance and angle encoding, f is the feature dimension, and $\max_c$ denotes the largest ternary angular embedding $r^A_{i,j,c}$ in the neighbourhood of the superpoint $\hat{p}_i$. Pair-wise distance embedding characterizes the proximity between point pairs, while triplet-wise angular embedding captures local shape information. By employing a learnable projection matrix, these distinct geometric features are mapped into a unified space, providing a comprehensive representation of the superpoint's geometric structure. This unified representation also facilitates subsequent processing by attention mechanisms.

In point cloud P, the *rotation invariant embedding* between each superpoint and other superpoints is denoted as $R^{\hat{P}}$. Then, $R^{\hat{P}}$ is concatenated with the superpoint feature $\hat{F}^P$ extracted by the backbone network:

$$F^{\hat{P}} = \mathrm{R}^{\hat{P}} \oplus \hat{F}^P \tag{7}$$

where $\oplus$ denotes concatenation along the feature dimension. The same operation is applied to point cloud Q. The concatenated features $F^{\hat{P}} \in \mathbb{R}^{|\hat{P}| \times d}$ and $F^{\hat{Q}} \in \mathbb{R}^{|\hat{Q}| \times d}$ are then fed into a Transformer composed of self-attention and cross-attention layers for further encoding. In the Transformer, the attention mechanism is defined as:

$$\mathrm{MHAtten}\left(\mathrm{Q}, \mathrm{K}, \mathrm{V}\right) = \left(\mathrm{head}_1 \oplus \ldots \oplus \mathrm{head}_H\right) \mathrm{W}^O \tag{8}$$

$$\mathrm{head}_h = \mathrm{Atten}\left(\mathrm{QW}_h^Q, \mathrm{KW}_h^K, \mathrm{VW}_h^V\right) \tag{9}$$

where $\oplus$ denotes concatenation along the channel dimension, and $\mathrm{W}_h^Q, \mathrm{W}_h^K, \mathrm{W}_h^V \in \mathbb{R}^{d \times d_{\mathrm{head}}}$ as well as $\mathrm{W}^O \in \mathbb{R}^{H d_{\mathrm{head}} \times d}$ are learnable projection matrices. The number of attention heads is set to $H = 4$, with $d_{\mathrm{head}} = d/H$. The feature interaction in each projection space is computed using the scaled dot-product attention:

$$\mathrm{Atten}\left(\mathrm{Q}, \mathrm{K}, \mathrm{V}\right) = \mathrm{softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{d_{head}}}\right) \mathrm{V} \tag{10}$$

In the self-attention layer, the query, key, and value matrices are set as $\mathrm{Q} = \mathrm{K} = \mathrm{V} = F^{\hat{P}}$. Each superpoint updates its feature representation based on its relationship with other superpoints, enabling intra-superpoint feature interaction and information propagation. In the cross-attention layer, the query is set as $\mathrm{Q} = F^{\hat{P}}$, $\mathrm{K} = \mathrm{V} = F^{\hat{Q}}$. This cross-attention mechanism facilitates feature interaction between the two point clouds, thereby enhancing their alignment and registration. Finally, the features of the two point clouds after the self-attention and cross-attention layers of the Transformer are represented as $\phi^{\hat{P}} \in \mathbb{R}^{|\hat{P}| \times d}$ and $\phi^{\hat{Q}} \in \mathbb{R}^{|\hat{Q}| \times d}$, respectively.

Hybrid similarity fusion module

Previous methods for calculating similarity often fail to consider interactions between features or overlook similarity within local regions of the point cloud. The HarSoNet network architecture evaluates the similarity between superpoint features using the proposed hybrid similarity fusion module. As shown in Fig. 4, the similarity between two superpoint features is initially calculated using Scaled Dot-Product Similarity (SDPS) and Pairwise Distance Similarity (PDS). The two similarity results are then combined using a 1×1 convolution. Scaled Dot-Product Similarity captures the global structure and feature interactions within the point cloud, while Pairwise Distance Similarity effectively describes spatial distribution and local feature similarities. Combining both methods provides a holistic understanding of the point cloud, enabling a more comprehensive assessment of feature similarity from local to global levels.

$\mathrm{SDPS} \in \mathbb{R}^{|\hat{P}| \times |\hat{Q}|}$ can be expressed as follows:

$$\mathrm{SDPS} = \mathrm{e}^{\left(-\frac{\phi^{\hat{P}} \cdot \left(\phi^{\hat{Q}}\right)^{\top}}{\sqrt{d}}\right)} \tag{11}$$

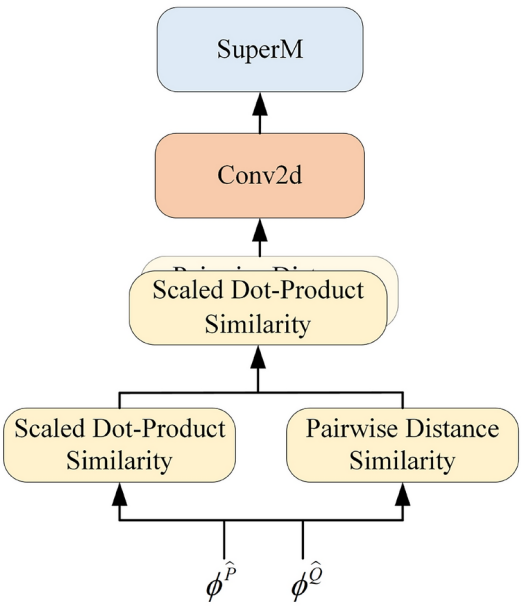

Fig. 4. Hybrid similarity fusion module.

where $\top$ denotes the transpose of the matrix. Similar to the attention mechanism in Transformer, SDPS computes the dot product between two feature vectors. The denominator $\sqrt{d}$ is used to scale the dot product, preventing similarity values from becoming excessively large when dealing with high-dimensional feature vectors. Since SDPS is based on the dot product operation, it effectively captures the global structure of the point cloud and facilitates feature interactions, enhancing the overall registration process. $\mathrm{PDS} \in \mathbb{R}^{|\hat{P}|\times|\hat{Q}|}$ can be expressed as follows:

$$\mathrm{PDS} = \mathrm{e}^{\left(-\left\|\phi^{\hat{P}}-\phi^{\hat{Q}}\right\|_2^2\right)} \tag{12}$$

PDS computes the squared Euclidean distance between two feature vectors and transforms it into a similarity measure using an exponential function. This approach effectively captures the spatial distribution of features, emphasizing how features are positioned relative to each other in the embedding space. By leveraging Euclidean distance, PDS provides a more geometrically intuitive similarity measure compared to SDPS. After stacking the two similarity results, a hybrid similarity matrix $C_{i,:,:} \in \mathbb{R}^{2\times|\hat{P}|\times|\hat{Q}|}$ can be expressed as:

$$C_{i,:,:} = \begin{cases} \mathrm{SDPS}, i = 0 \\ \mathrm{PDS}, i = 1 \end{cases} \tag{13}$$

The similarity matrices from different channels of $C_{i,:,:}$ are fused using a convolutional neural network, resulting in the final output $\mathrm{SuperM} \in \mathbb{R}^{|\hat{P}|\times|\hat{Q}|}$:

$$\mathrm{SuperM} = \mu_\theta\left(C_{i,:,:}\right) \tag{14}$$

where μ_θ denotes a two-dimensional convolutional layer. The combined similarity result leverages the advantages of SDPS, which is sensitive to the global structure of the point cloud, and PDS, which captures distance relationships. By integrating these two complementary measures, the final similarity computation becomes more robust, effectively balancing structural awareness and spatial sensitivity to improve matching accuracy in point cloud registration. Then, we apply double normalization to the SuperMmatrix[19] for suppressing anomalous matching:

$$\overline{\mathrm{SuperM}}_{i,j} = \frac{\mathrm{SuperM}_{i,j}}{\sum_{k=1}^{|\hat{Q}|}\mathrm{SuperM}_{i,k}} \cdot \frac{\mathrm{SuperM}_{i,j}}{\sum_{k=1}^{|\hat{P}|}\mathrm{SuperM}_{k,j}} \tag{15}$$

where $\mathrm{SuperM}_{i,j}$ is an element of the SuperM matrix. By applying dual normalization, ambiguous matches are suppressed, and mutually consistent reliable correspondences are selected, enhancing the distinctiveness of the matching process. The top N_c elements $\overline{\mathrm{SuperM}}_{x,y}$ in the normalized matrix $\overline{\mathrm{SuperM}}$ are selected as the hard matching correspondences between the superpoints:

$$\hat{C} = \left\{\left(\hat{p}_{x_i}, \hat{q}_{y_i}\right) \mid (x_i, y_i) \in \mathrm{topk}_{x,y}\left(\overline{\mathrm{SuperM}}_{x,y}\right)\right\} \tag{16}$$

Using Eq. (16), we obtain the set of correspondences between superpoints, denoted as $\hat{C}$. Each correspondence $\left(\hat{p}_{x_i}, \hat{q}_{y_i}\right)$ represents a matched pair of superpoints, while (x_i, y_i) denote the indices of $\overline{\mathrm{SuperM}}_{x,y}$.

Soft matching stage

The superpoint correspondences established in the hard matching stage provide crucial initial information and constraints for the soft matching stage. This enables the soft matching process to focus its search within a targeted region rather than the entire point cloud space, resulting in faster and more robust matching.

As shown in Fig. 5, each superpoint corresponds $\hat{C}_i = \left(\hat{p}_{x_i}, \hat{q}_{y_i}\right)$ aggregates nearby dense points using Eq. (2) and Eq. (3) to form the patch correspondences $\left(M_{x_i}^P, M_{y_i}^Q\right)$ and their corresponding features $\left(F_{x_i}^P, F_{y_i}^Q\right)$. The similarity matrix for each patch correspondence, denoted as PointM_i, can be expressed as:

$$\mathrm{PointM}_i = \mathrm{F}_{x_i}^P \left(\mathrm{F}_{y_i}^Q\right)^\top \Big/ \sqrt{\tilde{d}} \tag{17}$$

Here, $\top$ denotes the matrix transpose, and $\tilde{d}$ represents the feature dimension of the patch. Each element in PointM_i represents the similarity between each dense point $\tilde{p}$ in the local patch $M_{x_i}^P$ and each dense point $\tilde{q}$ in the local patch $M_{y_i}^Q$.

Sort-regenerate

Next, adjustments are made to PointM_i and $M_{x_i}^P$. Each row of the similarity matrix PointM_i represents the similarity between each dense point $\tilde{p}$ in the patch $M_{x_i}^P$ and each dense point $\tilde{q}$ in the patch $M_{y_i}^Q$. First, the maximum value in each row is selected, representing the similarity of each $\tilde{p}$ to its most similar $\tilde{q}$. Based on these results, the order of PointM_i and $M_{x_i}^P$ is adjusted to obtain the new similarity matrix $\mathrm{SortPointM}_i$ and the reordered source point cloud matrix $SortM_{x_i}^P$, facilitating optimal correspondence matching with the target point cloud:

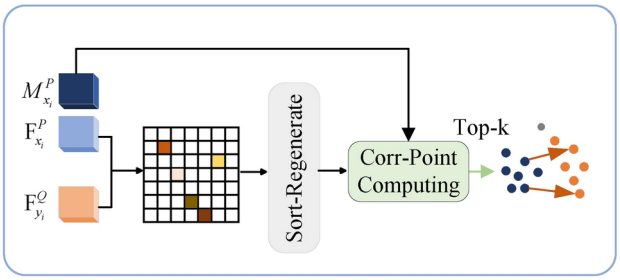

Fig. 5. Soft matching stage.

$$\mathrm{SortPoint}M_i = \mathrm{Point}M_i \left[\mathrm{argsort}(\max_k (\mathrm{PointM}_i)_{j,k})\right] \tag{18}$$

$$SortM_{x_i}^P = M_{x_i}^P \left[argsort(\max_k (\mathrm{PointM}_i)_{j,k})\right] \tag{19}$$

Corr-point computing
Finally, for each reordered source point cloud matrix $SortM_{x_i}^P$, the corresponding reference point cloud matrix $\bar{M}_{y_i}^Q$ can be expressed as:

$$\bar{M}_{y_i}^Q = \left(SortM_{x_i}^P\right)^\top \mathrm{SortPointM}_i \tag{20}$$

The top-k correspondences with the highest confidence scores from $(SortM_{x_i}^P, \bar{M}_{y_i}^Q)$ are chosen as the final selected point correspondences $\tilde{C}_i$:

$$\widetilde{C}_i = \mathrm{topk}\left(SortM_{x_i}^P, \bar{M}_{y_i}^Q\right) \tag{21}$$

Estimation of transformation parameters

All local point correspondences $\tilde{C}_i$ are aggregated into the global point correspondences $\tilde{C}$ using (17), where N_c represents the number of patch correspondences. The rigid transformation T is computed using (18), where the similarity scores from each soft matching correspondence $(\tilde{p}_{x_j}, \tilde{q}_{y_j})$ serve as the weights w_j:

$$\tilde{C} = \mathrm{Concat}(\widetilde{C}_i, i = 1, 2, \cdots, N_c) \tag{22}$$

$$\mathrm{R,t} = \min_{\mathrm{R,t}} \sum_{\left(\tilde{p}_{x_j}, \tilde{q}_{y_j}\right) \in \tilde{C}} w_j \left\|\mathrm{R} \cdot \tilde{p}_{x_j} + \mathrm{t} - \tilde{q}_{y_j}\right\|_2^2 \tag{23}$$

Loss function

The loss function $L = L_{oa} + L_{rt}$ in this paper comprises two components: the superpoint matching loss L_{oa} and the transformation parameters loss L_{rt}.

The superpoint matching process is supervised using the overlapping perceptual circle loss. A set of anchor patches Λ is defined with the local patch M_i^P of the source point cloud P and M_i^Q of the reference point cloud Q. Patches are categorised as positive pairs if they are repeatable; otherwise, they are classified as negative pairs. For each $M_i^P \in \Lambda$, the sets of positive and negative patches in Q are denoted as ω_t^i and ω_f^i, respectively. The overlapping perceptual circle loss on the source point cloud P is expressed as:

$$L_{oa}^P = \frac{1}{|\Lambda|} \sum_{M_i^P \in \Lambda} \log \left[1 + \sum_{M_j^Q \in \omega_t^i} \mathrm{e}^{\lambda_i^j \delta_t^{i,j} (d_i^j - \Delta_t)} \cdot \sum_{M_k^Q \in \omega_f^i} \mathrm{e}^{\delta_f^{i,k} (\Delta_f - d_i^k)}\right] \tag{24}$$

where $d_i^j = ||\phi_i^{\hat{P}} - \phi_j^{\hat{Q}}||_2$ is the Euclidean distance between the superpoint features, and $\lambda_i^j = (o_i^j)^{1/2}$ represents the overlap rate between the patches M_i^P and M_j^Q; futher details are provided in[16]. The hyperparameters $\Delta_t = 0.1$, $\Delta_f = 1.4$, and the weights of the positive and negative patch sets are defined as $\delta_t^{i,j} = \gamma(d_i^j - \Delta_t)$ and $\delta_f^{i,k} = \gamma(\Delta_f - d_i^k)$, representing the non-negative results of $(d_i^j - \Delta_t)$ and $(\Delta_f - d_i^k)$. The total superpoint matching loss is the average of the losses from point clouds P and Q: $L_{oa} = (L_{oa}^P + L_{oa}^Q)/2$.

To optimize rotation and translation parameters, reducing registration errors, the following loss function calculates the deviation between predicted and true transformation parameters:

$$L_{rt} = \lambda(\left\|\mathrm{R}^\mathrm{T} \cdot \hat{\mathrm{R}} - I\right\|^2 + \left\|\mathrm{t} - \hat{\mathrm{t}}\right\|^2) \tag{25}$$

where $\hat{\mathrm{R}}$ and $\hat{\mathrm{t}}$ are the true transformation parameters, R and t are the predicted transformation parameters, and λ is a balancing weight for the loss components.

Experiments

Implementation details

The experiments in this study were conducted on a system equipped with a 12-vCPU Intel® Xeon® Platinum 8352 V CPU running at 2.10 GHz and an NVIDIA RTX 3090 GPU. The computation framework is implemented using PyTorch, with 200 training rounds on the ModelNet40 dataset and 60 rounds on the 3DMatch dataset. The Adam optimizer is used for parameter updates. The batch size was set to 1. The initial learning rate was 0.0001, with a decay rate of 0.000001. During training, $N_g = 128$ real superpoint correspondences were used. In the testing phase, $N_c = 256$ estimated superpoint correspondences were employed.

ModelNet40 dataset

ModelNet40 comprises 40 categories of 3D CAD models. Eight symmetric categories (e.g., cups, vases) were excluded from both training and testing. The remaining categories were split into training (4,194 models), validation (1,002 models), and test (1,146 models) sets. Following[12], all points were randomly projected, and 70% ($p = 0.7$) were retained to create partially overlapping source and reference point clouds. The source point cloud was subjected to a rotational transformation within the range $[0, r = 45]$ and a translational transformation within $[-0.5, 0.5]$. The point cloud was then sampled twice: data from the first sampling was denoted as *OS*, and data from the second as *TS*.

To validate the performance of the proposed algorithm, we compare it with traditional methods, including ICP (point-to-point) and RANSAC, as well as learning-based methods such as DCP[10], RPM-NET[12], OMNet[41], and GeoTrans[16]on Unseen Shape, Unseen Categories and Gaussian Noises. We employ the implementations of ICP and RANSAC from Intel Open3D[42]and utilize (Fast Point Feature Histogram)FPFH[43] for feature matching. Three metrics were used to evaluate the transformation parameters: root mean square error (RMSE), mean absolute error (MAE), and isotropic error (Error).

Unseen shapes

In this study, we initially trained the model using full-class data and tested it on unseen *OS* and *TS* datasets. The results in Table 1 indicate that traditional algorithms perform worse across all metrics compared to learning-based methods, demonstrating the strong generalization ability and adaptability of learning-based approaches. The proposed HarSoNet outperforms other methods across all metrics. The Error(R) and Error(t) values are 0.846 and 0.007 for *OS*, and 1.575 and 0.011 for *TS*, respectively. Fig. 6 presents a qualitative comparison of learning-based methods on *OS*, where HarSoNet achieves the most accurate registration results, closely matching the ground truth.

Unseen categories

To assess the model's generalization ability, we trained it on the first 14 categories and tested it on the remaining unseen categories. The results are presented in Table 2. In this part of the experiment, the results of learning-based methods were generally inferior to their performance on Unseen Shape. Traditional algorithms demonstrated a certain level of robustness. HarSoNet achieved the best results on all *OS* data, with RMSE(R) and MAE(R) reaching 0.751 and 0.637, respectively. On *TS*data, its performance was second only to GeoTrans[16]. HarSoNet demonstrated strong robustness on unseen categories point cloud data. Fig. 7 presents a qualitative comparison of learning-based methods on *OS*.

Gaussian noise

To assess the models' robustness to noise, Gaussian noise of $N(0, 0.01^2)$ was added to the Unseen Categories, and each point was clipped to the range $[-0.05, 0.05]$. The experimental results, presented in Table 3, indicate that after adding Gaussian noise, all models exhibited some robustness. The proposed method achieved the best results across all *OS* metrics, with Error(R) reaching 1.376 and Error(t) 0.015, demonstrating strong noise resilience. Fig. 8 presents a qualitative comparison of learning-based methods on *OS*.

Method	RMSE(R)		MAE(R)		RMSE(t)		MAE(t)		Error(R)		Error(t)	
	OS	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*
RANSAC+FPFH	42.802	45.731	17.015	19.402	0.352	0.352	0.100	0.169	27.51	31.42	0.319	0.358
ICP	23.196	24.298	10.512	11.160	0.328	0.331	0.155	0.152	20.964	22.452	0.314	0.345
DCP	6.980	8.441	5.430	8.006	0.064	0.078	0.049	0.047	12.135	12.790	0.112	0.135
RPMNET	1.138	1.556	0.929	1.255	0.014	0.015	0.013	0.014	2.266	2.467	0.024	0.025
OMNet	0.857	1.261	0.781	1.142	0.007	0.011	0.006	0.009	1.454	2.066	0.012	0.018
GeoTrans	0.507	1.008	0.433	0.862	0.007	0.011	0.006	0.009	0.855	1.697	0.012	0.018
Ours	**0.489**	**0.909**	**0.427**	**0.581**	**0.004**	**0.007**	**0.004**	**0.006**	**0.846**	**1.575**	**0.007**	**0.011**

Table 1. Comparison of registration results on Unseen Shapes. **Boldfaced** numbers highlight the best and the second best are underlined.

(a) input (b) GT (c) DCP (d) RPMNET (e) OMNet (f) GeoTrans (g) Ours

Fig. 6. Qualitative examples of different network models on Unseen Shapes. (**a**) input. (**b**) Ground-truth(GT). (**c**)-(**g**) The qualitative examples with different network models.

Method	RMSE(R)		MAE(R)		RMSE(t)		MAE(t)		Error(R)		Error(t)	
	OS	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*
RANSAC+FPFH	46.115	50.650	19.472	23.166	0.388	0.412	0.174	0.191	31.903	37.610	0.362	0.416
ICP	23.376	24.545	10.702	11.702	0.325	0.336	0.154	0.168	21.113	23.288	0.338	0.341
DCP	9.582	10.403	7.650	8.399	0.088	0.104	0.060	0.103	16.290	17.981	0.153	0.180
RPMNET	4.055	7.036	3.942	5.381	0.018	0.030	0.017	0.024	7.421	12.236	0.032	0.052
OMNet	3.816	4.021	3.380	3.852	0.018	0.027	0.017	0.023	6.653	6.890	0.031	0.048
GeoTrans	0.972	**1.152**	0.826	**0.984**	0.010	**0.012**	0.008	**0.010**	1.631	**1.939**	0.017	**0.021**
Ours	**0.751**	1.909	**0.637**	1.639	**0.008**	0.023	**0.007**	0.018	**1.253**	3.187	**0.014**	0.039

Table 2. Comparison of registration results on Unseen Categories. **Boldfaced** numbers highlight the best and the second best are underlined.

(a) input (b) GT (c) DCP (d) RPMNET (e) OMNet (f) GeoTrans (g) Ours

Fig. 7. Qualitative examples of different network models on Unseen Categories. (**a**) input. (**b**) Ground-truth(GT). (**c**)-(**g**) The qualitative examples with different network models.

Method	RMSE(R)		MAE(R)		RMSE(t)		MAE(t)		Error(R)		Error(t)	
	OS	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*	*OS*	*TS*
RANSAC+FPFH	53.256	58.853	25.511	30.034	0.442	0.494	0.228	0.251	42.220	49.126	0.478	0.557
ICP	24.242	25.196	11.000	12.154	0.315	0.342	0.152	0.177	22.144	24.225	0.334	0.361
DCP	9.201	11.316	8.354	8.675	0.109	0.117	0.087	0.100	15.761	19.400	0.189	0.202
RPMNET	4.675	8.821	4.199	7.034	0.055	0.104	0.050	0.085	8.108	15.276	0.095	0.180
OMNet	4.143	4.570	3.905	4.328	0.030	0.037	0.023	0.029	6.464	6.672	0.052	0.064
GeoTrans	1.030	**1.152**	0.879	**0.991**	0.011	**0.013**	0.009	**0.011**	1.736	**1.940**	0.019	**0.022**
Ours	**0.820**	2.139	**0.705**	1.835	**0.009**	0.024	**0.007**	0.020	**1.376**	3.527	**0.015**	0.042

Table 3. Comparison of registration results on Gaussian Noise. **Boldfaced** numbers highlight the best and the second best are underlined.

Large rotation

In this part of the experiment, *OS* samples in the Gaussian Noise experiment were set to ModelNet with $p = 0.7$ and ModelLoNet with $p = 0.5$. The maximum rotation angle r between the two point clouds was set to 180°. Due to this setup, algorithms such as ICP and DCP[10]did not produce reasonable results; therefore, their outcomes are omitted from this paper. Meanwhile, Predator[14]and CoFiNet[15], which use similar correspondence-based methods, were included in the comparison experiments. The results of Predator[14]and CoFiNet[15]were refined

(a) input (b) GT (c) DCP (d) RPMNET (e) OMNet (f) GeoTrans (g) Ours

Fig. 8. Qualitative examples of different network models on Gaussian Noise. (**a**) input. (**b**) Ground-truth(GT). (**c**)-(**g**) The qualitative examples with different network models.

Method	ModelNet			ModelLoNet		
	RRE	RTE	CD	RRE	RTE	CD
RPMNET	31.682	0.211	0.01277	52.243	0.353	0.02048
Predator	20.524	0.152	0.01142	41.575	0.346	0.04896
CoFiNet	10.367	0.081	0.00316	32.899	0.234	0.02354
GeoTrans	6.218	0.043	0.00125	23.138	**0.145**	0.01192
Ours	**5.849**	**0.041**	**0.00104**	**22.822**	0.148	**0.01189**

Table 4. Comparison of registration results on Large Rotation. **Boldfaced** numbers highlight the best and the second best are underlined.

#Samples	3DMatch					3DLoMatch					Average
	5000	2500	1000	500	250	5000	2500	1000	500	250	Times(s)
D3 Feat	85.3	84.5	82.6	81.8	77.9	43.4	42.6	46.6	43.5	39.1	0.289
SpinNet	87.5	86.2	85.1	82.7	70.2	55.4	54.2	47.8	40.6	26.8	90.804
Predator	90.6	89.3	89.9	88.7	86.6	62.8	61.4	61.2	60.5	58.1	0.759
YOHO	90.8	90.3	89.1	88.6	84.5	65.2	65.2	63.2	56.5	48.0	13.529
GeoTrans	**92.0**	91.8	91.8	91.5	91.2	**75.2**	**75.0**	**74.7**	**74.3**	**73.5**	0.192
REGTR	**92.0**					64.8					0.382
Ours	91.8					67.5					**0.143**

Table 5. RR for each method across different numbers of correspondences. **Boldfaced** numbers highlight the best and the second best are underlined.

using RANSAC estimation. To clearly and intuitively evaluate differences in transformation parameters, the relative rotation error (RRE), the relative translation error (RTE), and corrected Chamfer Distance (CD)[12] were used. Table 4 presents the experimental results. The performance of all methods decreases when the initial poses of the two point clouds differ significantly. Performance further declines when dealing with point clouds that have low overlap. However, because hard-to-soft matching enhances the soft matching search space and establishes accurate correspondences, the proposed method outperforms others on most metrics.

3DMatch dataset

The 3DMatch dataset, comprising 62 3D scans of indoor scenes, is widely used for research in point cloud alignment, scene reconstruction, and robot navigation. The dataset includes 46 scenes for training, 8 for validation, and 8 for testing. Following the pre-processing protocol of Predator[14], the data are categorized into 3DMatch (overlap>30%) and 3DLoMatch (overlap between 10% and 30%).

To validate the performance of the proposed algorithms, comparisons were conducted with D3 Feat[11], SpinNet[21], Predator[14], YOHO[23], GeoTrans[16], and REGTR[40]on the 3DMatch and 3DLoMatch datasets. Registration Recall (RR) was used as the evaluation metric, and the inference time of all methods was recorded. Given the limited point correspondences generated by soft matching, the proposed algorithms were evaluated directly on all points, following the sparse matching approach of REGTR[40]. As shown in Table 5, the proposed algorithm achieves a RR of 91.8% on the 3DMatch dataset, ranking second only to GEGTR[40]. On the 3DLoMatch dataset, which has a low overlap rate, the algorithm attains a 67.5% RR, second only to GeoTrans[16]. Notably, the algorithm demonstrates the fastest runtime among all methods while maintaining sub-optimal RR, highlighting the efficiency of the two-stage combined soft and hard matching approach. Fig. 9 illustrates the registration results achieved by the proposed algorithm on the 3DMatch and 3DLoMatch datasets.

Private data

Fig. 10 presents the point cloud data for a launch vehicle engine diaphragm featuring a pressure-sensitive structure. The data contains approximately 200 million points. The point distributions are as follows: X-axis: [1.855, 82.325], Y-axis: [0.005, 89.995], and Z-axis: [0.355, 2.046].

In this part of the experiment, the original point cloud data is first scaled by a factor of 0.05. Following[12], all points are randomly projected, and points with a retention probability of $p = 1.0$ are kept to generate FullData, which represents the fully overlapping source and reference point clouds. Points with a retention probability of $p = 0.7$ are kept to generate LoData, representing partially overlapping source and reference point clouds. Then, FullData and LoData are randomly downsampled to retain approximately 35,000 points. Finally, rotation transformations within the range of $[0, r = 45]$ and translation transformations within the range of $[-0.5, 0.5]$ are applied to the reference point cloud in both experimental settings.

To evaluate the efficacy of our method, comparative benchmarking was conducted against conventional approaches, including ICP and RANSAC, as well as learning-based end-to-end methods such as RPM-NET[12]and GeoTrans[16], with experimental validation performed on FullData and LoData. All methods used training weights from the ModelNet40 dataset. Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Isotropic Error (Error) served as evaluation metrics. The results, shown in Table 6 and visualized in Fig. 11, indicate that the proposed algorithm outperforms others across all metrics. The suboptimal performance of

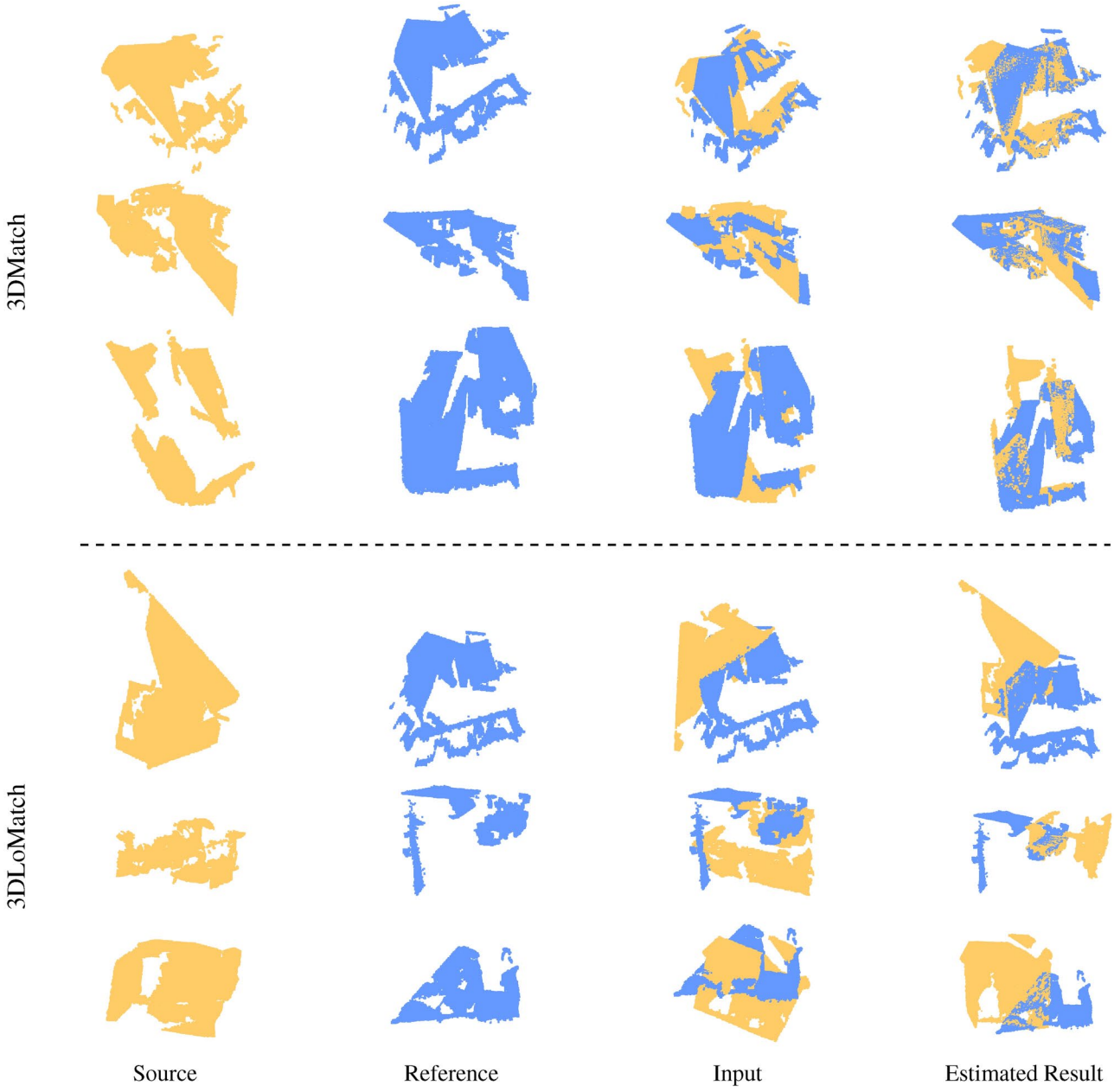

Fig. 9. Registration visualization on 3DMatch and 3DLoMatch.

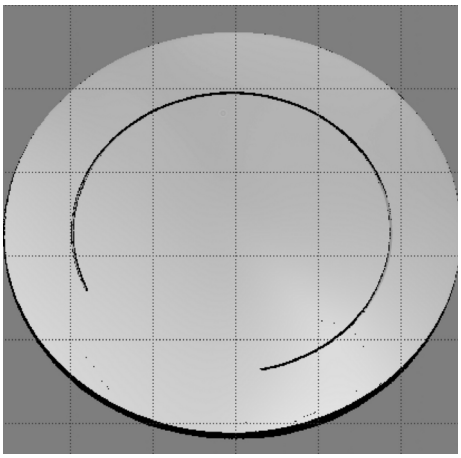

Fig. 10. Point cloud data of the original diaphragm.

Method	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	Error(R)	Error(t)
FullData						
RANSAC+FPFH	39.826	5.502	0.168	0.348	10.134	0.710
ICP	28.728	4.484	0.036	0.158	8.915	0.331
RPM-NET	0.898	0.743	0.018	0.015	6.074	0.031
GeoTrans	0.254	0.194	0.011	0.008	0.421	0.019
Ours	**0.118**	**0.088**	**0.001**	**0.001**	**0.194**	**0.002**
LoData						
RANSAC+FPFH	58.265	7.401	0.113	0.288	11.360	0.582
ICP	52.180	6.070	0.535	0.488	12.522	1.267
RPM-NET	0.920	0.842	0.015	0.011	6.308	0.027
GeoTrans	0.287	0.193	0.012	0.008	0.497	0.020
Ours	**0.157**	**0.106**	**0.008**	**0.006**	**0.274**	**0.014**

Table 6. Comparison of registration results on FullData and LoData. **Boldfaced** numbers highlight the best and the second best are underlined.

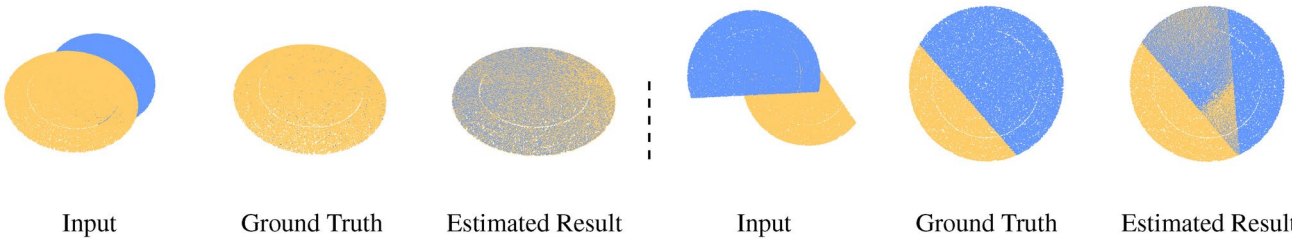

Fig. 11. Registration visualization on FullData and LoData. The left shows the visualization results of FullData, and the right shows the visualization results of LoData.

conventional point cloud registration approaches can be primarily attributed to the computational challenges posed by the sheer volume of data requiring alignment. A secondary contributing factor emerges in RANSAC-based implementations, where the absence of surface normal information in raw point clouds necessitates normal estimation during FPFH computation, introducing additional computational uncertainties that may propagate through subsequent registration stages. The visualization results show that the proposed algorithm aligns the diaphragm incision accurately, highlighting the method's effectiveness. Overall, these results highlight the algorithm's strong generalization ability on unseen data.

	SDPS	PDS	HTH	HTS	Error(R)	Error(t)
1)	✓			✓	3.411	0.041
2)		✓		✓	3.041	0.035
3)	✓	✓	✓		1.736	0.019
4)	✓	✓		✓	**1.376**	**0.015**

Table 7. Ablation of different modules on ModelNet40 dataset. **Boldfaced** numbers highlight the best.

Ablation studies

To assess the effectiveness of each module in the proposed model, we conducted experiments to evaluate different configurations, as summarized in Table 7. The experiment was conducted on the ModelNet40 dataset, following the data settings in Gaussian Noise, and using isotropic errors Error(R) and Error(t) for evaluation.

Calculation of similarity

This paper verifies the effectiveness of the proposed hybrid similarity fusion module. As shown in rows 1) and 2) of Table 7, using only scaled dot product similarity or pairwise distance similarity results in the worst registration performance. In contrast, combining both methods, as demonstrated in rows 3) and 4) of Table 7, achieves the best performance. Specifically, under the two-stage Hard-to-Hard (HTH) structure, Error(R) reaches 1.736 and Error(t) 0.019. In the Hard-to-Soft (HTS) structure, Error(R) improves to 1.376, and Error(t) reduces to 0.015.

Two-stage network structure

The hard-to-soft matching method forms the core structure of the proposed model. When compared with the two-stage hard matching method used in CoFiNet[15]and GeoTrans[16], results in rows 3) and 4) of Table 7 show that the HTS structure yields the lowest error. This indicates that the proposed hard and soft matching approach effectively reduces anomalous matches, thereby enhancing matching accuracy.

Conclusion

This paper proposes an end-to-end point cloud registration model that integrates two-stage hard and soft matching. In the hard matching phase, initial correspondences are quickly established through localization and coarse matching, while the hybrid similarity fusion module reduces anomalous matches and enhances reliability. The soft matching phase refines these correspondences, optimizing the coarse matches into more accurate point registrations and improving overall stability. By leveraging the speed of hard matching and the robustness of soft matching, the combined approach ensures efficient and reliable correspondences. Experimental results demonstrate that the proposed method is noise-robust and generalizes well to point clouds with unknown categories and shapes.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 21 January 2025; Accepted: 9 April 2025
Published online: 22 April 2025

References

1. He, Y.-R., Chen, P., Ma, W.-W. & Chen, C.-C. Construction of 3D Model of Tunnel Based on 3D Laser and Tilt Photography. *Sensors and Materials* **32**, 1743–1755 (2020).
2. Liu, F., Yang, B., Yang, Y., Zhao, Y. & Zhai, X. Space-constrained Mobile Laser-point Cloud Data Acquisition Method. *Sensors and Materials* **35**, 929–940 (2023).
3. Wang, J., Gong, Z., Tao, B. & Yin, Z. A 3-D Reconstruction Method for Large Freeform Surfaces Based on Mobile Robotic Measurement and Global Optimization. *IEEE Transactions on Instrumentation and Measurement* **71**, 1–9 (2022).
4. Wang, X. et al. High-precision point cloud registration system of multi-view industrial self-similar workpiece based on super-point space guidance. *Journal of Intelligent Manufacturing* **35**, 1765–1779 (2024).
5. Besl, P. & McKay, N. D. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 239–256 (1992).
6. Biber, P. & Strasser, W. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, 2743–2748 (IEEE, Las Vegas, Nevada, USA, 2003).
7. Zhou, Q.-Y., Park, J. spsampsps Koltun, V. Fast Global Registration. In Leibe, B., Matas, J., Sebe, N. spsampsps Welling, M. (eds.) *Computer Vision – ECCV 2016*, vol. 9906, 766–782 (Springer International Publishing, Cham, 2016).
8. Deng, H., Birdal, T. & Ilic, S. PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 195–205 (IEEE, Salt Lake City, UT, 2018).
9. Gojcic, Z., Zhou, C., Wegner, J. D. & Wieser, A. The Perfect Match: 3D Point Cloud Matching With Smoothed Densities. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5540–5549 (IEEE, Long Beach, CA, USA, 2019).
10. Wang, Y. & Solomon, J. Deep Closest Point: Learning Representations for Point Cloud Registration. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3522–3531 (IEEE, Seoul, Korea (South), 2019).
11. Bai, X. *et al.* D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6358–6366 (IEEE, Seattle, WA, USA, 2020).
12. Yew, Z. J. & Lee, G. H. RPM-Net: Robust Point Matching Using Learned Features. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11821–11830 (IEEE, Seattle, WA, USA, 2020).

13. Pan, L., Cai, Z. & Liu, Z. Robust partial-to-partial point cloud registration in a full range. *arXiv preprint* arXiv:2111.15606 (2021).
14. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A. & Schindler, K. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4265–4274 (IEEE, Nashville, TN, USA, 2021).
15. Yu, H., Li, F., Saleh, M., Busam, B. & Ilic, S. CoFiNet: Reliable Coarse-to-fine Correspondencess forRobust Point Cloud Registration. *Advances in Neural Information Processing Systems* **34**, 23872–23884 (2021).
16. Qin, Z. *et al.* Geometric Transformer for Fast and Robust Point Cloud Registration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11133–11142 (IEEE, New Orleans, LA, USA, 2022).
17. Li, X., Han, K., Li, S. & Prisacariu, V. Dual-Resolution Correspondence Networks. *Advances in Neural Information Processing Systems* **33**, 17346–17357 (2020).
18. Zhou, Q., Sattler, T. & Leal-Taixe, L. Patch2Pix: Epipolar-Guided Pixel-Level Correspondences. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4667–4676 (IEEE, Nashville, TN, USA, 2021).
19. Sun, J., Shen, Z., Wang, Y., Bao, H. & Zhou, X. LoFTR: Detector-Free Local Feature Matching with Transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8918–8927 (IEEE, Nashville, TN, USA, 2021).
20. Lu, F. *et al.* HRegNet: A Hierarchical Network for Large-scale Outdoor LiDAR Point Cloud Registration. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 15994–16003 (IEEE, Montreal, QC, Canada, 2021).
21. Ao, S., Hu, Q., Yang, B., Markham, A. & Guo, Y. SpinNet: Learning a General Surface Descriptor for 3D Point Cloud Registration. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11748–11757 (IEEE, Nashville, TN, USA, 2021).
22. Yang, F., Guo, L., Chen, Z. & Tao, W. One-Inlier is First: Towards Efficient Position Encoding for Point Cloud Registration. *Advances in Neural Information Processing Systems* **35**, 6982–6995 (2022).
23. 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*, 1630–1641 (ACM, Lisboa Portugal, 2022).
24. Poiesi, F. & Boscaini, D. Learning general and distinctive 3D local deep descriptors for point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**, 3979–3985 (2022).
25. Wang, H. et al. RoReg: Pairwise Point Cloud Registration With Oriented Descriptors and Local Rotations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**, 10376–10393 (2023).
26. Tang, K. et al. Manifold Constraints for Imperceptible Adversarial Attacks on Point Clouds. *Proceedings of the AAAI Conference on Artificial Intelligence* **38**, 5127–5135 (2024).
27. Peng, W. et al. Deep Correspondence Matching-Based Robust Point Cloud Registration of Profiled Parts. *IEEE Transactions on Industrial Informatics* **20**, 2129–2143 (2024).
28. Wang, S., Tong, Y. & Zhang, Z. Multi-Constraints Guided Single-View Point Cloud Registration for Adaptive Robotic Manipulation. *IEEE Transactions on Industrial Electronics* 1–11 (2025).
29. Pais, G. D. *et al.* 3DRegNet: A Deep Neural Network for 3D Point Registration. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7191–7201 (IEEE, Seattle, WA, USA, 2020).
30. Lee, J., Kim, S., Cho, M. & Park, J. Deep Hough Voting for Robust Global Registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15994–16003 (2021).
31. Zhang, Y.-X., Sun, Z.-L., Zeng, Z. & Lam, K.-M. Partial Point Cloud Registration With Deep Local Feature. *IEEE Transactions on Artificial Intelligence* **4**, 1317–1327 (2023).
32. Bai, X. *et al.* PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15854–15864 (IEEE, Nashville, TN, USA, 2021).
33. Chen, Z., Sun, K., Yang, F. & Tao, W. SC ^{2} -PCR: A Second Order Spatial Compatibility for Efficient and Robust Point Cloud Registration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13211–13221 (IEEE, New Orleans, LA, USA, 2022).
34. Zhang, X., Yang, J., Zhang, S. & Zhang, Y. 3D Registration with Maximal Cliques. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17745–17754 (IEEE, Vancouver, BC, Canada, 2023).
35. Tang, K. *et al.* SymAttack: Symmetry-aware Imperceptible Adversarial Attacks on 3D Point Clouds. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 3131–3140 (ACM, Melbourne VIC Australia, 2024).
36. Chen, Z., Yang, F. & Tao, W. DeTarNet: Decoupling Translation and Rotation by Siamese Network for Point Cloud Registration. *Proceedings of the AAAI Conference on Artificial Intelligence* **36**, 401–409 (2022).
37. Xu, H., Ye, N., Liu, G., Zeng, B. & Liu, S. FINet: Dual Branches Feature Interaction for Partial-to-Partial Point Cloud Registration. *Proceedings of the AAAI Conference on Artificial Intelligence* **36**, 2848–2856 (2022).
38. Thomas, H. *et al.* KPConv: Flexible and Deformable Convolution for Point Clouds. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6410–6419 (IEEE, Seoul, Korea (South), 2019).
39. Lin, T.-Y. *et al.* Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944 (IEEE, Honolulu, HI, 2017).
40. Yew, Z. J. & Lee, G. H. REGTR: End-to-end Point Cloud Correspondences with Transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6667–6676 (IEEE, New Orleans, LA, USA, 2022).
41. Xu, H., Liu, S., Wang, G., Liu, G. & Zeng, B. OMNet: Learning Overlapping Mask for Partial-to-Partial Point Cloud Registration. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3112–3121 (IEEE, Montreal, QC, Canada, 2021).
42. Zhou, Q.-Y., Park, J. & Koltun, V. Open3D: A Modern Library for 3D Data Processing (2018). arXiv:1801.09847.
43. Rusu, R. B., Blodow, N. & Beetz, M. Fast Point Feature Histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation*, 3212–3217 (IEEE, Kobe, 2009).

Acknowledgements

This work was supported by National Key Research and Development Program of China under Grant 2022YFB4601700.

Author contributions

Q.H. managed the project, acquired the fund and revised the manuscript; J.W. performed the experiments, wrote and revised the manuscript; J.H. conceived the work, designed the experiments, guided and revised the manuscript; S.K. performed the investigation.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Q.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.