Supplementary Materials for Equi-GSPR: Equivariant SE(3) Graph Network Model for Sparse Point Cloud Registration

Xueyang Kang[1,2,3], Zhaoliang Luan[2,4], Kourosh Khoshelham[3], and Bing Wang[2]*

[1] Faculty of Electrical Engineering, KU Leuven
[2] Spatial Intelligence Group, The Hong Kong Polytechnic University
[3] Faculty of Engineering and IT, The University of Melbourne
[4] IoTUS Lab, Queen Mary University of London
alex.kang@kuleuven.com, z.luan@qmul.ac.uk, k.khoshelham@unimelb.edu.au, bingwang@polyu.edu.hk

1 Equivariant Graph Network Model

We provide a detailed view of the proposed model structure for transforming the stacked $N \times (32 + 3)$ tensors from the sparsely sampled input points, including the respective feature shape dimensions as indicated below.

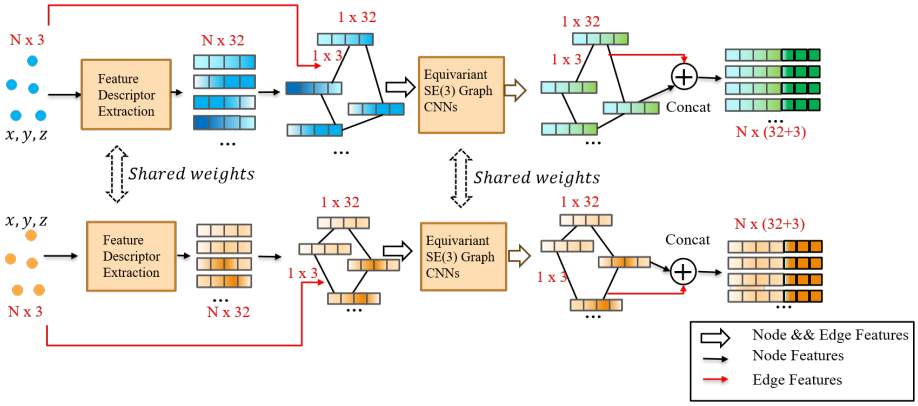

Fig. 1: Initially, the process involves sparse input points from two frames, followed by the extraction of point-wise feature descriptors. Subsequently, a graph is constructed based on these descriptor features. The feature descriptor graphs then pass through individual equi-graph layers. The resulting graph node features are combined with the average edge embedding features to form tensors of dimensions $N \times (32 + 3)$ for the decoder.

The descriptor is generated point-wise from the sparsely sampled points in both the source and target frames. Subsequently, this descriptor, combined with

* Corresponding author

the $\mathbf{SO}(3)$ edge coordinate embeddings discussed in the main body document, is used to create the feature graph. This graph is then inputted into graph convolution network layers. Following the equivariant graph layers, the resulting aggregated graph node features, combined with the average of neighbouring coordinate feature embeddings, are reorganized into a 2D tensor sized $N \times (32+3)$ before entering the Low-Rank Feature Transformation (LRFT) module.

1.1 Graph Equivariance Proof

Let's start by clarifying what equivariance means. Consider a graph $G = (V, E)$ with node features represented as $\mathbf{x}_v \in \mathbb{R}^d$ for each node $v \in V$, and a collection of transformations $\mathcal{G}$ that act on the graph. A Graph Neural Network (GNN) layer is considered equivariant to $\mathcal{G}$ if it meets the following criterion:

$$\mathbf{h}'_v = \rho_{\mathcal{G}}((g \cdot \mathbf{x}_v, g \cdot \mathbf{e}_{u \to v}) \mid u \in \mathcal{N}(v)) \tag{1}$$

$$= g \cdot \rho_{\mathcal{G}}(\mathbf{x}_u, \mathbf{e}_{u \to v} \mid u \in \mathcal{N}(v)) \tag{2}$$

where $\mathbf{h}'_v$ represents the updated node feature for node v, $\rho_{\mathcal{G}}$ denotes the equivariant graph convolution operation, $\mathcal{N}(v)$ indicates the set of neighbours of node v, $\mathbf{e}_{u \to v}$ is the edge feature from node u to node v, and $g \in \mathcal{G}$ represents a group transformation that operates on the node and edge features. To ensure equivariance, it is essential that the graph convolution operation $\rho_{\mathcal{G}}$ is formulated to preserve the group structure of $\mathcal{G}$. For instance, if $\mathcal{G}$ represents the permutation group that influences the node indices, $\rho_{\mathcal{G}}$ needs to remain unchanged when nodes are permuted.

Now, let's further discuss the concept of invariance. A Graph Neural Network (GNN) model is considered invariant to $\mathcal{G}$ if its final output, such as the graph prediction of a node, remains unchanged when subjected to group transformations $\mathcal{G}$. This can be mathematically represented as:

$$\mathbf{y} = \rho_{\text{inv}}(\mathbf{h}_v \mid v \in V) = \rho_{\text{inv}}(g \cdot \mathbf{h}_v \mid v \in V) \tag{3}$$

where $\mathbf{y}$ is the final output, ρ_{inv} is an invariant pooling operation (e.g., sum, max, or invariant multi-head attention), and $\mathbf{h}_v \mid v \in V$ are the node representations obtained after applying equivariant graph CNN layers.

To maintain invariance, the pooling operation ρ_{inv} needs to be constructed in a way that remains unchanged when subjected to group transformations within $\mathcal{G}$. For instance, if $\mathcal{G}$ represents the permutation group, ρ_{inv} should exhibit invariance towards permutations of nodes. By combining equivariance and invariance, a Graph Neural Network (GNN) can be formulated as follows,

Apply equivariant GNN layers to update node representations:

$$\mathbf{h}_v^{(l+1)} = \rho_{\mathcal{G}}^{(l)}(\mathbf{h}_u^{(l)}, \mathbf{e}_{u \to v} \mid u \in \mathcal{N}(v)) \tag{4}$$

Apply an invariant pooling operation to obtain the final output:

$$\mathbf{y} = \rho_{\text{inv}}(\mathbf{h}_v^{(L)} \mid v \in V)$$

where l is the layer index, and L is the total number of GNN layers.

This framework enables Graph Neural Networks (GNNs) to utilize group symmetries existing in the data, enhancing the efficiency and robustness of the learning process. The accurate configurations of the equivariant graph convolution operations denoted as $\rho_{\mathcal{G}}^{(l)}$ and the invariant pooling operation denoted as ρ_{inv} are determined by the selected group $\mathcal{G}$ and the architecture of the GNN, here we use the sum and pooling operation to learn the invariance.

1.2 Matrix Multiplication Rank Theorem Proof

Theorem 1. Let $\boldsymbol{A}$ be an $N \times r$ matrix, and $\boldsymbol{B}$ be an $r \times N'$ matrix. We want to prove that:

$$\text{Rank}(\boldsymbol{AB}) \leq \min(\text{Rank}(\boldsymbol{A}), \text{Rank}(\boldsymbol{B})) \tag{5}$$

The proof relies on the following concepts, the rank of a matrix is equal to the dimension of its column space:

Proof: Let $\mathbf{A}$ be an $N \times r$ matrix and $\mathbf{B}$ be an $r \times N'$ matrix. The rank of a matrix is the maximum number of linearly independent columns (or rows) in the matrix. Let us denote the columns of $\mathbf{A}$ as $[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r]$, and the rows of $\mathbf{B}$ as $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r]^T$. The results of $\mathbf{AB}$ are dot products of the columns of $\mathbf{A}$ and the rows of $\mathbf{B}$. Therefore, the maximum number of linearly independent columns in $\mathbf{AB}$ is bounded by the minimum of the number of linearly independent columns in matrix $\mathbf{A}$ or independent rows of $\mathbf{B}$.

To be noted, When the ratio of inliers to outliers in the feature correspondence is low, it can cause the feature similarity score matrix $\hat{\boldsymbol{S}}$ in main body of paper to become rank-deficient with rank $r \ll 35$. This can result in difficulty for the training loss to converge due to uncertainty in the feature space. To tackle this issue, we adopt an iterative approach to seek a viable full-rank solution r' that is smaller than r. This approach aims to minimize the overall training loss by ensuring that the full-rank condition of the submatrix is satisfied. The minimum value for the rank r' is set at 16; any rank lower than this threshold may lead to a significant increase in registration errors through our experiments, consequently causing the registration process to fail.

2 t-SNE Visualization of Equivariant Feature

As illustrated in Fig. 2, The process includes mapping the graph feature through Low-Rank constrained MLP layers onto the input point coordinates.

Initially, a feature similarity search is conducted by computing the dot product between feature vectors arranged along the row dimension, both pre- and post-LRFT, resulting in matrix shapes of $N \times 35$ and $N' \times 35$ respectively. The similarity matrix is then obtained through the Kronecker product, yielding an $N \times N'$ matrix. An argmax operation is applied to each row to identify the most similar pre-LRFT feature descriptor. The resulting index retrieves the corresponding input point coordinate, preserving feature descriptor association

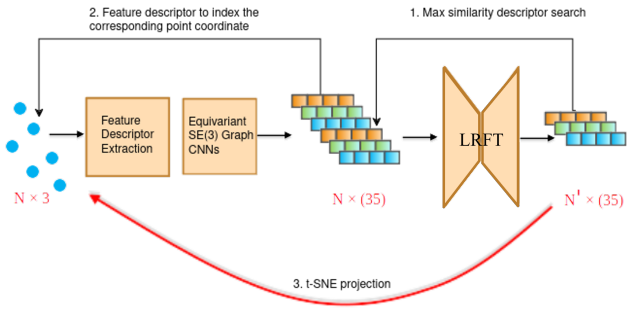

Fig. 2: The pipeline uses t-SNE [6] to map the sparse feature after the Low-Rank Feature Transformation (LRFT) module into the color map, then superimposed with the input points for visualization.

at the point cloud level. Notably, equi-graph layers maintain the original number and sequence of input points. These mapping stages establish a link between post-LRFT feature vectors and input points. Finally, 35-dimensional feature vectors are transformed into scalar color values using t-SNE [6] and superimposed onto input point coordinates for visualization. Fig. 3 displays the descriptor feature output from the pipeline illustrated in Fig. 2. These features correspond to scalar values associated with input point coordinates of the initial input scan.

The mapped features are uniformly distributed throughout the scan, with notable concentrations along edges and corners (as evident on furniture surfaces in the first and second rows of Fig. 3) (zoomed-in for better view). This visualization demonstrates that the equivariant features, post-equivariant graph layers, effectively represent distinctive geometric features in the input scan points. Furthermore, comparing the left and right column results reveals similar color and positional distributions of feature points between source and target frames, indicating a favorable correspondence distribution.

3 Experiment Results

We present additional visual comparison results of our model against baseline models, focusing on the top three quantitative performers from the main paper. For 3DMatch, the best baseline models include **SpinNet** [1], **RoReg** [7], and **PointDSC** [2]. For the KITTI dataset, we compare our model with **DGR** [4], **SpinNet** [1], and **PointDSC** [2], which represent the top three baseline models.

Our proposed model demonstrates robust registration capabilities and high accuracy across diverse scenarios, contrasting with some learning models that exhibit performance degradation when transitioning from indoor to outdoor. This discrepancy is particularly evident in PointDSC's performance gap between 3DMatch (1st column of Fig. 4) and KITTI (2nd column), where registration failure occurs in the initial KITTI case due to significant rotation errors.

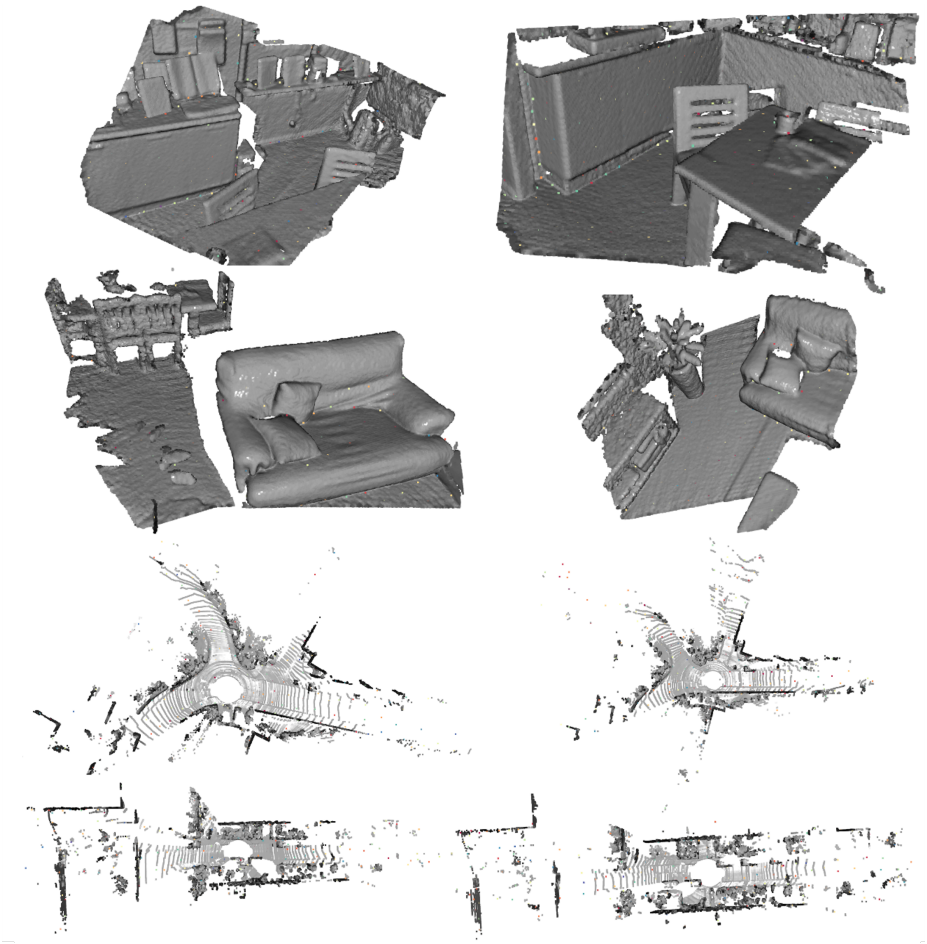

Fig. 3: The left and right sides represent the source and target frame outcomes, respectively. Colored points (zoomed-in for better view) indicate mapped t-SNE feature values, while grey meshes depict raw input scans for visualization. The feature mapping pipeline employs t-SNE [6] to map the output descriptors of post-Low-Rank Feature Transformation into a scalar color map. These are then superimposed onto respective input points from source and target frames to facilitate visualization.

3.1 Generalization on Unseen Datasets

To demonstrate the robust generalization capabilities of our graph-based representation, we conduct a generalization test by directly evaluating the 3DMatch pre-trained model on the KITTI dataset, as detailed in Tab. 1. Each evaluation approach was repeated 10 times to report average performance and standard deviation of errors. Notably, our model consistently achieves a high registration recall rate of 82.31%, accompanied by minimal rotation and translation errors.

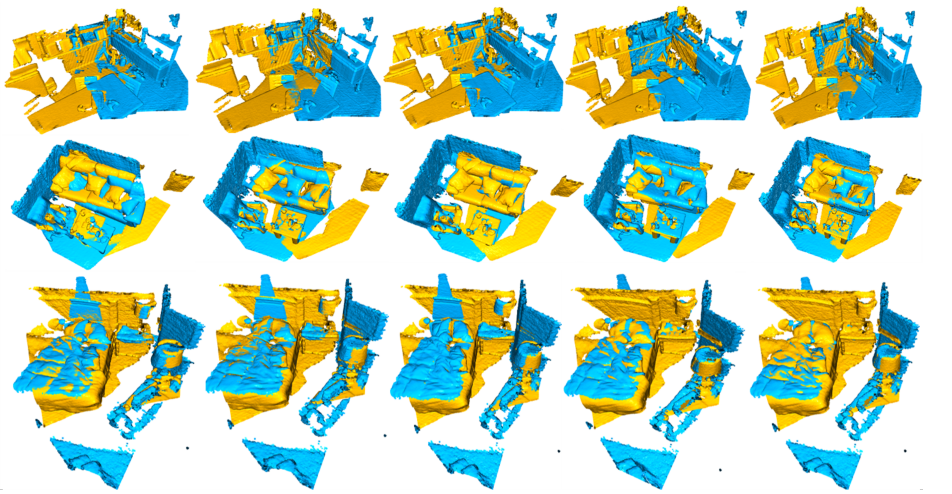

Fig. 4: Visual comparisons on 3DMatch, the three models with top performance in the main paper are presented. Points from the target frame are represented in blue, whereas points converted from the source frame by the predicted transform are depicted in yellow.

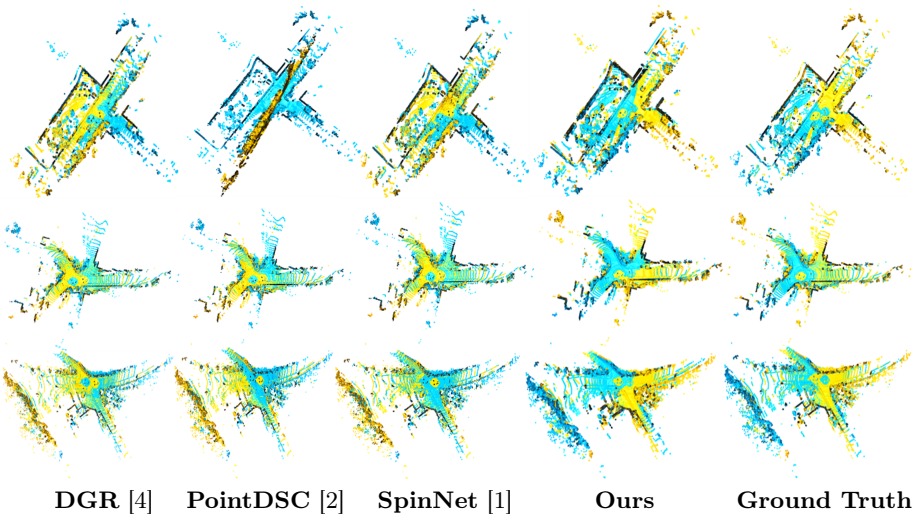

Fig. 5: Visual comparison results on KITTI, the three models with top performance in the main paper are selected for visualization. Points from the target frame are colored in blue, whereas points converted from the source frame by the predicted transform are illustrated in yellow.

Table 1: All the models are pre-trained on 3DMatch, and tested directly on KITTI.

	RE(°) ↓		TE(cm) ↓		RR(%) ↑
	AVG	STD	AVG	STD	
FCGF [5]	1.61	1.51	27.1	5.58	24.19
D3Feat(rand) [3]	1.44	1.35	31.6	10.1	36.76
SpinNet [1]	0.98	**0.63**	15.6	1.89	81.44
Ours	**0.86**	0.68	**10.7**	**1.23**	**82.31**

4 Ablation Study

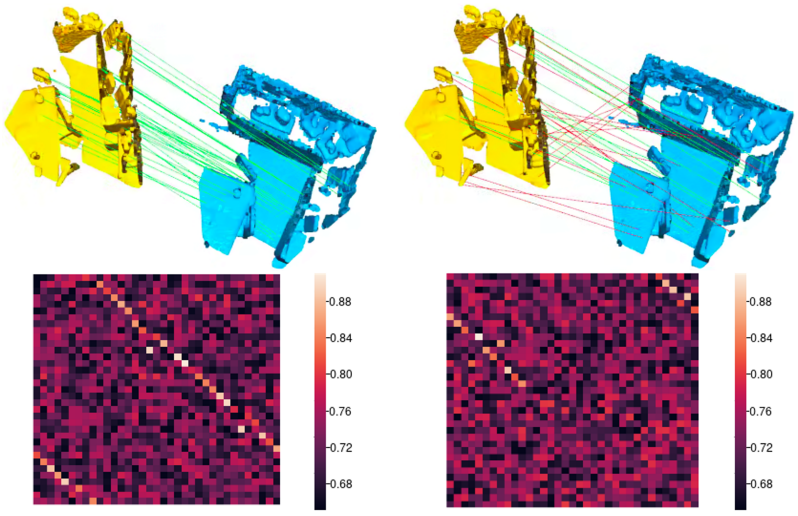

Fig. 6: This figure illustrates the comparison between correspondence results with and w/o equivariant features in graph layers horizontally, and the relationship vertically between the feature similarity score matrix and point correspondence. The top row displays the visualization of point correspondence linked to feature similarity below.

To demonstrate the relationship between the rank of the feature similarity score matrix $\hat{\boldsymbol{S}}$ and point correspondence, we specifically examine the model with normal graph CNN layers or equi-graph CNN layers. This analysis includes cases both with and without incorporating equivariance into graph layers to facilitate a comparative visualization of equivariance impact for registration. We extract the top 35 similarity score values from each case to identify pairwise features. These pairwise features are then linked with the feature descriptor before the LRFT module by selecting the maximum descriptor similarity in each row of similarity score matrix. Subsequently, this process allows us to retrieve the input point coordinates. By following these procedures, the feature pair after LRFT module can be correlated with the input point for visualization. The bottom row of Fig. 6 indicates that the adoption of equivariant features significantly enhances feature

distinctiveness. This enhancement facilitates the generation of valid, unique rank values within the similarity score matrix, as illustrated on the left side matrix rank. In contrast, the application of non-equivariant features tends to increase ambiguity in match score computations. Moreover, a binary indicator ω_i for visual assessment of point correspondences requires the comparison of distance errors against the inlier threshold τ as below,

$$\omega_i = \mathbb{1}[\|\hat{\boldsymbol{R}}\boldsymbol{x}_i + \hat{\boldsymbol{t}} - \boldsymbol{y}_j\| < \tau], \tag{6}$$

The label one is depicted as a green line while the zero is represented by the red line in the top row of Fig. 6.

Finally, we evaluated the impact of neighboring node count on feature graph initialization and accuracy performance, as measured by the RMSE metric (please refer to Fig. 7).

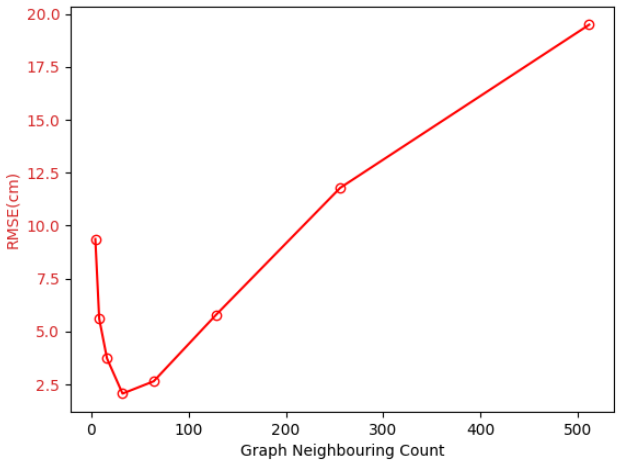

Fig. 7: RMSE plot as a function of graph neighbouring feature node count.

Performance improves as the number of neighboring nodes used for graph creation increases to 24. However, a significant performance deterioration occurs when the node count exceeds 200. This decline suggests that an excessive number of neighboring feature nodes can cause overflow of information for the whole graph feature learning, dispersing attention during feature aggregation and consequently reducing performance due to ambiguous neighboring features. While the plot displays a graph node count limit of 512, the actual limit is 1024, which unfortunately triggers out-of-memory issues during model computations in our hardware settings.

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. Bai, X., Luo, Z., Zhou, L., Chen, H., Li, L., Hu, Z., Fu, H., Tai, C.L.: Pointdsc: Robust point cloud registration using deep spatial consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15859–15869 (2021)
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. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2514–2523 (2020)
5. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
6. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
7. Wang, H., Liu, Y., Hu, Q., Wang, B., Chen, J., Dong, Z., Guo, Y., Wang, W., Yang, B.: Roreg: Pairwise point cloud registration with oriented descriptors and local rotations. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)