A Strong Baseline for Point Cloud Registration via Direct Superpoints Matching

Aniket Gupta[1], Yiming Xie[1], Huaizu Jiang[1], Hanumant Singh[2]

***Abstract*— Deep neural networks endow the downsampled superpoints with highly discriminative feature representations. Previous dominant point cloud registration approaches match these feature representations as the first step, *e.g.*, using the Sinkhorn algorithm. A RANSAC-like method is then usually adopted as a post-processing refinement to filter the outliers. Other dominant method is to directly predict the superpoint matchings using learned MLP layers. Both of them have drawbacks: RANSAC-based methods are computationally intensive and prediction-based methods suffer from outputing non-existing points in the point cloud. In this paper, we propose a straightforward and effective baseline to find correspondences of superpoints in a global matching manner. We employ the normalized matching scores as weights for each correspondence, allowing us to reject the outliers and further weigh the rest inliers when fitting the transformation matrix without relying on the cumbersome RANSAC. Moreover, the entire model can be trained in an end-to-end fashion, leading to better accuracy. Our simple yet effective baseline shows comparable or even better results than state-of-the-art methods on three datasets including ModelNet, 3DMatch, and KITTI. We do not advocate our approach to be *the* solution for point cloud registration but use the results to emphasize the role of matching strategy for point cloud registration. The code and models are available at https://github.com/neu-vi/Superpoints_Registration**

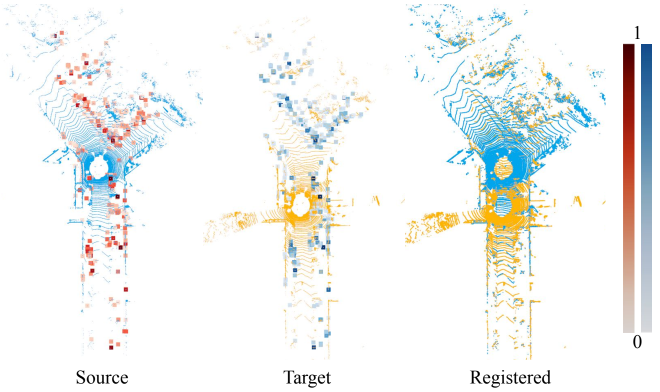

Fig. 1. **Our approach directly matches the superpoints between the input point clouds to register them by estimating the $SE(3)$ transformation matrix**. The correlation weights obtained from matching are used to filter out incorrect correspondences (outliers) and further weigh the rest inliers for the transformation matrix estimation. The source cloud is shown in blue, with superpoints highlighted in red. Similarly, the target point cloud is displayed in yellow, with matching superpoints shown in blue. The intensity of the red and blue colors represents the correspondence weights. Notice how most keypoints are distributed around the three-way junction and exhibit high correspondence weights. (Best viewed in color.)

I. Introduction

Point cloud registration refers to the task of aligning two partially overlapping point clouds into a shared coordinate system. In this paper, we tackle this problem where the goal is to determine the transformation matrix, including rotation and translation, from one point cloud (source) to the other (target). It has attracted a lot of research interest due to its broad applications in SLAM (Simultaneous Localization and Mapping) [14], [44], autonomous driving [28], [32], 3D reconstruction [22], [19], etc.

A prevailing paradigm to solve the registration task is to leverage the correspondences of *superpoints* across two point clouds, which can be obtained using either keypoint detectors [1], [4], [11], [21] or downsampling in deep neural networks [37], [38], [47]. Ideally, these superpoints should capture salient and distinctive points or regions within a point cloud. With the learned feature representations, the superpoints are endowed with sufficient discriminative power so they can be matched across the source and target point clouds, where the matching strategy is crucial for finding correspondences of superpoints and can significantly affect the convergence and performance of the network. Previous approaches [58], [39] use the Sinkhorn algorithm [45] to match superpoints, which however is sensitive to initialization parameters and requires careful tuning. In [46], Dual-Softmax is adopted for feature matching between two input images by using kepoints which have softmax correlation scores greater than a certain threshold. The correspondences of superpoints, however, are inevitably noisy due to either the noisy sensory data or simply incorrect estimations of correspondences, which prevent existing approaches from directly matching superpoints for point cloud registration. Postprocessing refinement is usually needed, for instance, using RANSAC-like approaches to prune the outliers [11], [12], [63], [61]. But such refinement is inherently slow due to the iterative nature of the RANSAC pipeline and can not be easily integrated into an end-to-end trainable system.

The recent work [59] populates another strategy, which uses a two-layer MLP to direct predict matched superpoints between the input point clouds. But such predicted correspondences may not exist in the point cloud, leading to inferior registration accuracy.

In this paper, we present a *strong baseline* for point cloud registration by directly matching superpoints. Specifically, building upon RegTR [59], instead of using a MLP to predict correspondences of superpoints, we find the correspondences by computing the similarity scores of all superpoints across the source and target point cloud in a global matching manner.

[1]Khoury College of Computer Science, Northeastern University
[2]College of Engineering, Northeastern University {gupta.anik, xie.yim, ha.singh, h.jiang}@northeastern.edu

Their normlized matching scores can be used to filter out the unreliable correspondences (*i.e.*, outliers). By integrating the weights of the rest inlier superpoint matchings into a differentiable variant of the Kabsh-Umeyama Algorithm [50], [23], we obtain robust estimations of the $SE(3)$ transformation between the input point cloud pairs. An illustration is shown in Fig. 1. As a result, no ad hoc postprocessing refinement is needed, yielding a more efficient model. More importantly, the entire model can be trained in an end-to-end manner, where the feature representation learning, superpoints matching, and transformation estimation can be jointly optimized. Better registration accuracy can thus be obtained. In addition, compared to [59], our approach find correspondences of superpoints by global matching, which does not output non-existing points in the target point cloud.

We run experiments on three benchmark datasets, including ModelNet [54], 3DMatch [63], and KITTI [16], and achieve comparable or even better results than state-of-the-art methods. We do not advocate our approach to be *the* solution for point cloud registration. Rather, we'd like to emphasize the role of matching strategy for point cloud registration, showcasing the feasibility of achieving high accuracy without cumbersome ad hoc postprocessing. By releasing the code and model weights, we hope our work can foster future research.

II. Related Work

Traditional registration approaches. The most known algorithm Iterative Closest Point (ICP) [5] has been widely used for point cloud registration. ICP solves the registration problem iteratively in two steps: (1) It obtains the spatially closest point correspondence and then (2) finds the least-squares rigid transformation. The spatial-distance-based correspondences are sensitive to the initial transformation and point noises. A lot of variants [41], [9], [43], [6], [40], [30] have been proposed to improve ICP. Another method is to extract and match keypoints based on feature extraction methods such as FPFH [42] and SHOT [48], followed by an outlier rejection postprocessing step.

Learning-based registration approaches. Recently, many works have used deep learning for point cloud learning and registration. Some work first estimates the correspondence between two point clouds and then computes the transformation with some robust pose estimators. To predict the correspondence between two point clouds, 3DMatch [63] detects the repeatable keypoints and learns discriminative descriptors for keypoints. The following works aim to either improve the keypoint detections [4], [26], [57] or learn better feature descriptors [11], [12], [13], [24], [1], [52]. Predator [21] uses the attention mechanism proposed in Transformers [51] to enhance the point feature descriptors. Other detector-free methods [61], [39] extract the correspondences by considering all possible matches. Another line of work [8], [62] has included the transformation computation into the training pipeline. Unlike these works which require either ad-hoc postprocessing or coarse-to-fine registration, our method directly matches the superpoints without any refinement.

Learning visual correspondence. Various approaches have been proposed to establish correspondences between the input, *e.g.*, using nearest neighbor followed by the distance ratio test [29]. Recent approaches [58], [39] use the Sinkhorn algorithm [45] to match superpoints for point cloud registration. However, it is sensitive to initialization parameters and requires careful tuning. In [46], Dual-Softmax is adopted for feature matching between two input images by using kepoints which have softmax correlation scores greater than a certain threshold. Global Softmax is used in a recent work of optical flow [56] to find global matchings by simply taking the correspondence for each pixel with the highest correlation score. In this paper, we extend Global Softmax by using the corrleation weights for both filtering unreliable correspondences and weighing the remaining inliers when estimating the SE(3) transformation matrix.

Correspondence filters. RANSAC [15] is typically used to filter out the outliers in the predicted correspondence to obtain a robust transformation estimation. However, RANSAC is relatively slow and cannot be incorporated into the training pipeline because the hypothesis selection step is non-differentiable. To alleviate these problems, DSAC [7] and ∇-RANSAC [53] modify the RANSAC pipeline and make it differentiable. But even the differentiable versions are similar in computational complexity as vanilla RANSAC. Other deep robust estimators [3], [10], [35], [17], [25], [60] usually use the classification network to identify which correspondences are outliers and then reject them. Instead of using these complex correspondence filters, our method can directly filter out outliers effectively by leveraging the rich information in the superpoints matching.

III. Method

Given the source and target point clouds $\mathbf{X} \in \mathbb{R}^{M\times 3}$ and $\mathbf{Y} \in \mathbb{R}^{N\times 3}$, our goal is to determine the SE(3) transformation $\mathbf{T} = \{\mathbf{R}, \mathbf{t}\}$ with rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ to align two point clouds into a common coordinate system. M and N denote the numbers of points.

A. Superpoints Feature Extraction and Enhancement

Following [59], we use Kernel Point Convolution (KPConv) [47] as the backbone to selectively downsample the point cloud into a set of superpoints and extract global feature vectors for each superpoint. The KPConv backbone uses a series of ResNet-like blocks [20] and convolutions to downsample the input point clouds into a reduced set of superpoints $\mathbf{X}' \in \mathbb{R}^{M'\times 3}$ and $\mathbf{Y}' \in \mathbb{R}^{N'\times 3}$, where $M' < M$ and $N' < N$. The superpoints are described by their feature vectors $\mathbf{F}_{\mathbf{X}'} \in \mathbb{R}^{M'\times D}$ and $\mathbf{F}_{\mathbf{Y}'} \in \mathbb{R}^{N'\times D}$, respectively, with D being the feature dimension. The network weights are shared among the two point clouds. We use a shallower backbone for 3DMatch dataset compared to [59], [21] to avoid significant downsampling by removing the 4-th residual block.

Although KPConv backbone provides reasonably good representations, the superpoints features are obtained within each point cloud *independently*. To obtain highly discriminative feature representations for superpoints matching, We

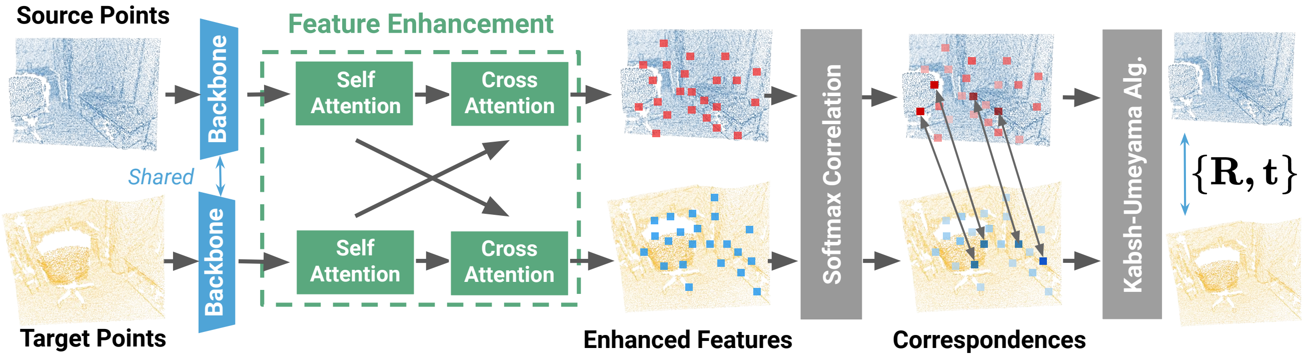

Fig. 2. **Model Architecture**: The KPConv backbone downsamples the input point cloud and generates superpoints and their feature vectors. These superpoint features are then conditioned on the other point cloud in the feature enhancement block. Lastly, superpoint features can be directly matched using Global Softmax to estimate SE(3) transformation using the corrleation weights in a robust manner. The intensity of the red and blue colors of the superpoints after the softmax correlation step represents the correspondence weights. (Best viewed in color.)

use the multi-head attention mechanism in the Transformer model [51] as the feature enhancement module as suggested in [59], shown in Fig. 2. It consists of both self and cross-attention, where the self-attention is to integrate the information from the other points within the same point cloud and the cross-attention allows interactions with points in another point cloud to consider the mutual dependencies. In addition to the multi-head attention, other components in the Transformer model, including position encodings of 3D points, residual connections, layer normalization, and feed-forward network are applied to each layer. The entire feature enhancement module consists of 6 such layers with 256 dimensions and 8 attention heads.

The outputs of the feature enhancement module are features $\bar{\mathbf{F}}_{\mathbf{X}'} \in \mathbb{R}^{M' \times D}$ and $\bar{\mathbf{F}}_{\mathbf{Y}'} \in \mathbb{R}^{N' \times D}$ which has aggregated geometric information from both source and target point cloud. The strongly associated features are strengthened while the weakly associated features are weakened.

B. Superpoint Matching for SE(3) Transformation Estimation

Unlike [59] that uses a two-layer MLP to predict corresponding points, we find the correspondences of superpoints in a global matching manner. This change removes the ambiguity of predicting non-existing corresponding points. To get the correspondences between two point clouds, we first compare the feature similarity for each point in $\bar{\mathbf{F}}_{\mathbf{X}'}$ to all points in $\bar{\mathbf{F}}_{\mathbf{Y}'}$ by computing their correlations [56], which can be done efficiently in a single step as follows:

$$\mathbf{C} = \text{Softmax}\left(\bar{\mathbf{F}}_{\mathbf{X}'} \cdot \bar{\mathbf{F}}_{\mathbf{Y}'}^{T}\right) \in \mathbb{R}^{M' \times N'}, \tag{1}$$

where $\mathbf{C}$ is the normalized correlation matrix representing the similarity between two point clouds. Based on the correlation matrix, the correspondences between $\mathbf{X}'$ and $\mathbf{Y}'$ can be directly calculated by using the largest correlation for each point.

The SE(3) transformation between the source and target point clouds can be then estimated using the superpoints correspondences with a weighted variant of the Kabsch-Umeyama algorithm [23], [50]:

$$\hat{\mathbf{R}}, \hat{\mathbf{t}} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i}^{\min(N',M')} w_i \|\mathbf{R}\hat{\mathbf{x}}_i + \mathbf{t} - \hat{\mathbf{y}}_i\|^2, \tag{2}$$

where $w_i = \mathbf{C}(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)$ is the correspondence weight, $\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}_i$ are the i-th pair of matched superpoints. We use $\min(N', M')$ operation because corresponding points can only be found for the point cloud with the minimum number of superpoints.

Although we use the highly discriminative feature representations enhanced by the attention module, the correspondences are inevitably noisy. How to filter out the outliers (*i.e.*, incorrect correspondences)? We show that the normalized correlation matrix $\mathbf{C}$ obtained from superpoints matching in Eq. (1) contains *rich information*, allowing us to effectively reject outliers for robust transformation matrix estimation. Specifically, if $\hat{\mathbf{x}}_i$ is similar to multiple superpoints, *e.g.*, because of the repetitive patterns, its matching to $\hat{\mathbf{y}}_i$ tends to be unreliable. Therefore, the normalized correlation score between them $w_i = \mathbf{C}(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)$ will be low since $\mathbf{C}$ is normalized w.r.t. all other superpoints in the target point cloud. Unlike [56], which completely discards the correlation weights in $\mathbf{C}$, we use w_i for two purposes: discarding the unreliable corresondences that have low w_i values and further weighting the rest inliers when fitting the transformation matrix in Eq. (1). It is important to note here that w_i is not learned and is just the correlation score between the feature representations of $\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}_i$, the i-th pair of matched superpoints. We show in the experiments that such a matching strategy works more effectively for superpoints matching than other approaches [49], [56], [46].

In comparison to [39], [61], [33], our approach is *simple yet effective*, which eliminates the coarse-to-fine strategy and more importantly, the inherently slow RANSAC-like postprocessing. Although the feature enhancement module is also used in [59], our approach is fundamentally different. [59] predicts corresponding points for both source and target point clouds whereas our approach *removes the ambiguity of predicting non-existing points* and directly matches superpoints. The limitation of [59] can also be seen in outdoor LiDAR experiments like KITTI. As shown in Table III, it performs significantly worse due to predicting correspondences that do not align well with true corresponding points.

C. Loss Functions

We train our approach using the following three loss functions, where the transformation loss is the main loss term and the other two are auxiliary ones.

Transformation Loss. We apply the L1 loss on the predicted transformed locations of all keypoints with the predicted and ground truth transformation matrix. This is different from [59]. Since we do not predict corresponding points for each point cloud, we do not need to compute two transformation matrices and thus we do not have to force the network to learn that both matrices are inverses of each other.

$$\mathscr{L}_T = \frac{1}{M'}\sum_{i}^{M'} \left|\hat{\mathbf{R}}\mathbf{x}'_{\mathbf{i}} + \hat{\mathbf{t}} - \left(\mathbf{R}_{gt}\mathbf{x}'_{\mathbf{i}} + \mathbf{t}_{gt}\right)\right|_1 . \tag{3}$$

Overlap Loss. Inspired by [59], we estimate the overlap values $\hat{\mathbf{O}}_{\mathbf{X}'}$ and $\hat{\mathbf{O}}_{\mathbf{Y}'}$ using a separate MLP layer based on the enhanced feature $\bar{\mathbf{F}}_{\mathbf{X}'}$ and $\bar{\mathbf{F}}_{\mathbf{Y}'}$, respectively. The overlap estimation is formulated as a binary classification problem, so we use the binary cross-entropy loss:

$$\mathscr{L}_o^X = -\frac{1}{M'}\sum_{i}^{M'} o^*_{x,i} \cdot \log \hat{o}_{x,i} + \left(1 - o^*_{x,i}\right) \cdot \log\left(1 - \hat{o}_{x,i}\right), \tag{4}$$

where $\hat{o}_{x,i}$ is the estimated overlap probability and $o^*_{x,i}$ is the ground truth probability. We compute the overlap loss $\mathscr{L}_o^Y$ for the target point cloud similarly.

Feature Loss. Following [59], to ensure that the enhanced features of both point clouds are in the same feature space, we apply an InfoNCE [34] loss on the enhanced features $\bar{\mathbf{F}}_{\mathbf{X}'}$ and $\bar{\mathbf{F}}_{\mathbf{Y}'}$. Given a set of superpoints correspondences $\left\{(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)\right\}_{i=1}^K$ and their associated feature representations $\left\{(\bar{\mathbf{f}}_{\hat{\mathbf{x}}_i}, \bar{\mathbf{f}}_{\hat{\mathbf{y}}_i}\right)$, the feature loss is defined as

$$\mathscr{L}_f = -\frac{1}{K}\log\frac{\bar{\mathbf{f}}_{\hat{\mathbf{x}}_i}^T \mathbf{W}\bar{\mathbf{f}}_{\hat{\mathbf{y}}_i}}{\bar{\mathbf{f}}_{\hat{\mathbf{x}}_i}^T \mathbf{W}\bar{\mathbf{f}}_{\hat{\mathbf{y}}_i} + \sum_{j\neq i}\bar{\mathbf{f}}_{\hat{\mathbf{x}}_i}^T \mathbf{W}\bar{\mathbf{f}}_{\hat{\mathbf{y}}_j}}. \tag{5}$$

The linear transformation $\mathbf{W}$ is enforced to be symmetrical by parameterizing it as the sum of an upper triangular matrix $\mathbf{U}$ and its transpose, *i.e.* $\mathbf{W} = \mathbf{U} + \mathbf{U}^T$.

The final loss is a weighted sum of all the losses with

$$\mathscr{L} = \mathscr{L}_T + \alpha\mathscr{L}_f + \beta(\mathscr{L}_o^X + \mathscr{L}_o^Y), \tag{6}$$

where we set the loss weights $\alpha = 0.1$ and $\beta = 1$ empirically.

IV. Experiments

We evaluate our approach on three datasets with overlap ranging from 10% to 75%. The first dataset is on synthetic ModelNet dataset with two benchmarks settings following [21], [59]. The second one is 3DMatch [63] with two benchmarks following [21], [59], [39], [61]. The last one is on the challenging large-scale outdoor KITTI dataset [16]

TABLE I

Registration results on ModelNet and ModelLoNet.

Methods	ModelNet			ModelLoNet		
	RRE↓	RTE↓	CD↓	RRE↓	RTE↓	CD↓
PNLK [2]	29.725	0.297	0.0235	48.567	0.507	0.0367
OMNet [55]	2.947	0.032	0.0015	6.517	0.129	0.0074
DCPv2 [31]	11.975	0.171	0.0117	16.501	0.300	0.0268
RPMNet [58]	1.712	0.018	0.00085	7.342	0.124	0.0050
Predator [21]	1.739	0.019	0.00089	5.235	0.132	0.0083
RegTR [59]	1.473	0.014	0.00078	3.930	**0.087**	**0.0037**
GeoT [39]	1.568	0.018	-	**3.809**	0.102	-
Ours	**1.247**	**0.011**	**0.00074**	**3.809**	0.088	0.0040

A. Implementation details

Our approach is implemented using the PyTorch framework [36] on a system with an Intel i9-1300K CPU and a single RTX 3090 GPU. The network training is performed with the AdamW optimizer [31], using a learning rate of 0.0001 and a weight decay of 0.0001. We tain the network for 400 epochs with batch size of 4 on ModelNet, 50 epochs with batch size of 4 on 3DMatch, and 80 epochs with batch size of 1 on KITTI.

B. ModelNet and ModelLoNet Benchmarks

The ModelNet40 [54] dataset comprises of synthetic CAD models. Following the data setting in [21], [59], the point clouds are randomly sampled from mesh faces of the CAD models, cropped and subsampled.

Our network is trained exclusively on ModelNet, and evaluated for generalization on ModelLoNet. For benchmarking the performance of our model we use the Relative Rotation Error (RRE) and Relative translation Error (RTE) and Chamfer Distance (CD) as the primary metrics, following [59].

The results are shown in Table I. We compare against correspondence-based approaches [21], [59], coarse-to-fine registration approaches [39], and end-to-end methods [2], [58], [55]. Our approach performs well on both benchmarks improving results on ModelNet benchmark by 15% in RRE and 21% in RTE. The low chamfer error suggests that predicted correspondences have very high accuracy. Our approach is also able to outperform methods using post-processing steps like RANSAC [21] by a significant margin.

C. 3DMatch and 3DLoMatch Benchmarks

3DMatch [63] is a collection of 62 scenes, from which we use 46 for training, 8 for testing, and 8 for validation following [59], [21], [39]. We use the preprocessed data from [21] which contains point clouds downsampled using a voxel-grid subsampling method. The 3DMatch benchmark contains point clouds pairs with >30% overlap while the 3DLoMatch benchmark contains scan pairs with only 10%-30% overlap. Following [59], we perform training data augmentation by applying small rigid perturbations, jittering, and shuffling of points. Following the literature [21], [39], [63], We report the results of 3DMatch dataset on 5 metrics including RRE, RTE, Registration Recall (RR), Feature Matching Recall (FMR), and Inlier Ratio (IR).

TABLE II

REGISTRATION RESULTS ON 3DMATCH AND 3DLOMATCH.

Methods	3DMatch					3DLoMatch				
	RRE↓	RTE↓	RR↑	FMR↑	IR↑	RRE↓	RTE↓	RR↑	FMR↑	IR↑
FCGF	1.949	0.066	85.1	97.4	56.8	3.147	0.100	40.1	76.6	21.4
D3Feat	2.161	0.067	81.6	95.6	39.0	3.361	0.103	37.2	67.3	13.2
Predator	2.029	0.064	89.0	96.6	58.8	3.048	0.093	59.8	78.6	26.7
DGR	2.103	0.067	85.3	-	-	3.954	0.113	48.7	-	-
RegTR	1.567	0.049	92.0	-	-	2.827	0.077	64.8	-	-
YOHO	-	-	90.8	**98.2**	64.4	-	-	65.2	79.4	25.9
CofiNet	-	-	89.3	98.1	49.8	-	-	67.5	83.1	24.4
GeoT	1.625	0.053	91.5	97.7	70.3	**2.547**	**0.074**	**74.0**	**88.1**	43.3
Ours	**1.436**	**0.045**	**93.7**	96.5	**89.8**	2.553	**0.074**	65.0	76.5	**57.5**

We compare our approach against several learned correspondence-based algorithms [18], [11], [4], [21] and coarse-to-fine approaches [61], [39]. The quantitative results are shown in Table II. For the 3DMatch benchmark, our approach outperforms all the previous methods in all but the FMR metric. This implies that in cases of significant overlap (>30%), the superpoint correspondences are very distinctive and accurate. For the 3DLoMatch benchmark, in comparison to all the approaches that do not use any post-processing, *e.g.* Predator[21] and CoFiNet[61], our method performs significantly better. GeoT [39] uses Local to Global Refinement (LGR) as the refinement step to get best results. In our case, outliers are filtered by the correlation weights, thus providing accurate correspondences. The validity of the correspondences obtained by our approach can be verified by comparing the mean Inlier Ratio (IR) in Table II. We get the highest mean IR, almost **19%** better than the second-best approach on 3DMatch and **14%** higher in 3DLoMatch. Our approach does not perform the best in FMR because we prioritize outputting correct correspondences over a large number of, yet potentially noisy matchings. In contrast, CofiNet [61], has a very high FMR but low IR due to its propensity for producing many inaccurate matches.

One of the problems associated with superpoint matching in point clouds is the resolution issue, where we might not have one-to-one correspondences due to subsampling. In our case, this is automatically handled by the correlation weights. The correlation weights are lower for weakly matching superpoints and thus even if there is no one-to-one correspondence, weak correspondences will be used according to their weights to compute the correct SE(3) transformation. Figure 3 shows the correlation weight distribution on the 3DMatch and 3DLoMatch benchmark. In 3DMatch, we have many high correlation weights, suggesting that many sample points are good correspondences. On 3DLoMatch, we have much fewer good correlations suggesting that many of the sampled superpoints are not good matches, which is expected because of very low overlap.

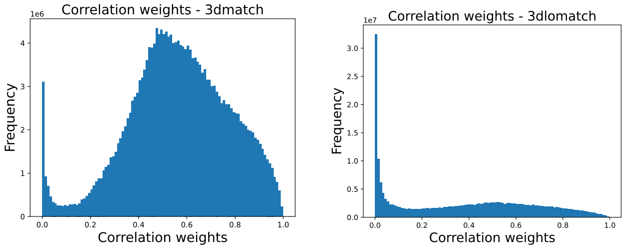

Fig. 3. **Correlation weight distribution on 3DMatch.** Here, many points exhibit moderate correlations, suggesting that a considerable portion of sample points are reasonably good matches. In contrast, 3DLoMatch has fewer points that exhibit strong correlations due to the low data overlap.

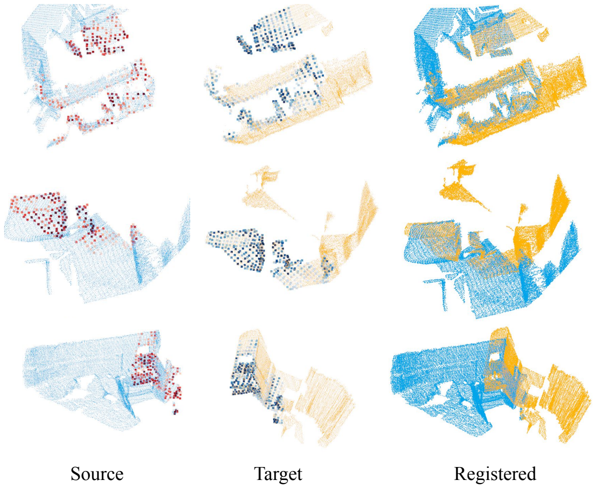

Fig. 4. **Qualitative Results on 3DMatch.** The source cloud is shown in blue with the red color representing the matched superpoints. Similarly target cloud is shown in yellow color with matching superpoints in blue. (Best viewed in color.)

D. KITTI Benchmark

KITTI odometry consists of 11 sequences of outdoor driving scenarios scanned using the HDL64 LiDAR sensor. Following [21], we use sequence 0-5 for training, 6-7 for validation, and 8-10 for testing. As in [21], [39] we only use the point cloud pairs that are at least 10m apart and their ground-truth poses are refined using ICP for all evaluations. Table III shows the results obtained on KITTI. We can see that the baseline [59] performs significantly worse in RTE. Since the correspondences are predicted, they do not align one-on-one with actual correspondences, thus significantly reducing registration accuracy. We achieve state of the art results in RRE and RR while performing comparably in RTE. This showcases the effectiveness of our approach in large scale outdoor point clouds. Figure 1 shows an example of matched superpoints on KITTI dataset, our approach is able to find very distinctive correspondences to register the point cloud pair.

E. Ablation Studies

Using correlation scores for filtering outliers. We study the effectiveness of outlier filtering in Table IV. In the 'No filtering' approach, we use all the superpoints with their scores *without any filtering*. A little surprisingly, it achieves very competitive results, likely because we use the correlation weights of the matchings in the transformation matrix estimation in Eq.(2). If the outlier filtering is used (the last row), the best accuracy can be obtained.

TABLE III

REGISTRATION RESULTS ON KITTI.

Methods	RTE↓	RRE↓	RR↑
3DFeat [57]	25.9	0.25	96.0
FCGF [42]	9.5	0.30	96.6
D3Feat [4]	7.2	0.30	**99.8**
Predator [21]	**6.8**	0.27	**99.8**
CofiNet [61]	8.2	0.41	**99.8**
RegTR [59]	49.9	0.61	99.1
GeoTransformer [39]	**6.8**	**0.24**	**99.8**
Unified BEV [27]	7.5	0.26	**99.8**
Ours	7.2	**0.24**	**99.8**

TABLE IV

COMPARISONS OF DIFFERENT OUTLIER FILTERING METHODS.

Outlier Filtering	3DMatch			3DLoMatch			
Method	RRE	RTE	RR	RRE	RTE	RR	Time
No filtering	1.462	0.046	93.4	2.652	**0.074**	64.6	**0.073**
Ours + RANSAC	2.788	0.090	82.3	4.653	0.115	32.0	0.141
Ours (top 15%) + RANSAC	1.701	0.051	93.0	2.696	0.078	64.5	0.153
Correlation scores (top 15%)	**1.436**	**0.045**	**93.7**	**2.553**	**0.074**	**65.0**	**0.073**

We also experiment with RANSAC on the set of superpoints correspondences with and without outlier filtering, respectively. Without any filtering (‘Ours + RANSAC’), the large number of outliers impose significant challenges for RANSAC to find the optimal solution. Even with filtering (‘Ours (top 15%) + RANSAC’), we can see that RANSAC gives slightly worse results, partially because it does not always converge to the optimal solution. Furthermore, RANSAC requires twice as much time for the estimation of the transformation matrix compared to our method.

Effectiveness of the loss terms. We also analyze the effectiveness of each loss function. Table V shows the results with different loss function configurations. We see that by just using the feature loss along with the transformation loss, the model is able to achieve good performance. Using the overlap loss further helps the model prune remaining outliers.

Matching strategy. The choice of a matching strategy is crucial for precise pose estimation. In prior approaches, such as [39], [58], [61], the Sinkhorn algorithm has commonly been employed to compute optimal transport and derive matching scores for correspondences. In our research, we opt for Global Softmax, which, as evidenced by our experiments, outperforms the Sinkhorn approach (refer to Table VI). Note that while Sinkhorn algorithm requires careful parameter tuning, our approach is parameter-free and works out of the box. While Global Softmax has been previously used as a matching strategy in [49], [58], prior works only utilized it to identify matching points with probabilities exceeding a certain threshold, resulting in suboptimal outcomes. Our innovation lies in utilizing these probabilities as correspondence weights, facilitating gradient flow through the Weighted Kabsch-Umeyama Solver which leads to faster optimization and better results. We also experimented with Dual Softmax matching used in LofTR [46], but the difference in performance is less than 1% with Global Softmax being slightly better and faster than Dual-Softmax.

TABLE V

EFFECTIVENESS OF DIFFERENT LOSS TERMS.

Overlap	Feature	3DMatch			3DLoMatch		
Loss	Loss	RRE	RTE	RR	RRE	RTE	RR
✗	✗	2.521	0.076	76.5	5.272	0.132	31.2
✓	✗	2.123	0.062	79.6	4.020	0.105	37.2
✗	✓	1.651	0.049	90.3	2.876	0.082	58.0
✓	✓	**1.436**	**0.045**	**93.7**	**2.553**	**0.074**	**65.0**

TABLE VI

COMPARISON OF MATCHING STRATEGY

Methods	3DMatch			3DLoMatch			Time
	RRE↓	RTE↓	RR↑	RRE↓	RTE↓	RR↑	
Sinkhorn	1.509	0.046	89.2	2.622	**0.074**	59.6	0.094
Dual Softmax	1.441	**0.045**	93.6	2.589	**0.074**	**65.1**	0.080
MLP [59]	1.567	0.049	92.0	2.827	0.077	64.8	0.078
Global Softmax (Ours)	**1.436**	**0.045**	**93.7**	**2.553**	**0.074**	65.0	**0.073**

TABLE VII

COMPARISON OF DATA AUGMENTATION SCHEMES

Methods	3DMatch			3DLoMatch		
	RRE↓	RTE↓	RR↑	RRE↓	RTE↓	RR↑
Ours (Weak Augmentation)	**1.436**	**0.045**	**93.7**	**2.553**	**0.074**	65.0
Ours (Large Augmentation)	1.472	**0.045**	93.4	2.611	**0.074**	**66.8**

Data Augmentation Schemes Previous works have tried different data augmentation schemes. [59] uses a weak data augmentation, perturbing poses by a small amount during training. While [39], [58], [21] use large data augmentation, perturbing poses by a full range of motion. Weak data augmentation can slightly improve performance but deteriorates network generalization. Table VII shows that our approach achieves slightly better performance with weak data augmentation. But with strong data augmentation, the generalization on 3DLoMatch benchmarks improves RR by 1.8%.

V. CONCLUSION

In this paper, we presented a strong baseline approach for point cloud registration. By using Global Softmax to directly match superpoint features, we remove the ambiguity of predicting non-existent corresponding points while using the softmax probabilities as correspondence weights allows us to filter outliers without any post-processing refinement. Experimental results on standard benchmarks show that our model achieves comparable or even better accuracy than state-of-the-art methods. We do not advocate our approach to be *the* solution for point cloud registration. Rather, we’d like to emphasize the role of matching strategy for point cloud registration, showcasing the feasibility of achieving high accuracy without cumbersome ad hoc postprocessing.

References

[1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *CVPR*, 2021.

[2] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *CVPR*, 2019.

[3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *CVPR*, 2021.

[4] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *CVPR*, 2020.

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, 1992.

[6] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse iterative closest point. In *Computer graphics forum*, 2013.

[7] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017.

[8] Anh-Quan Cao, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Pcam: Product of cross-attention matrices for rigid registration of point clouds. In *ICCV*, 2021.

[9] Dmitry Chetverikov, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek. The trimmed iterative closest point algorithm. In *ICPR*, 2002.

[10] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, 2020.

[11] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *CVPR*, 2019.

[12] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *ECCV*, 2018.

[13] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *CVPR*, 2018.

[14] Jean-Emmanuel Deschaud. IMLS-SLAM: scan-to-model matching based on 3d data. In *ICRA*, 2018.

[15] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[17] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *CVPR*, 2020.

[18] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *CVPR*, 2019.

[19] Johannes Groß, Aljoša Ošep, and Bastian Leibe. Alignnet-3d: Fast point cloud registration of partially observed objects. In *3DV*, 2019.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021.

[22] Shahram Izadi, Richard A. Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J. Davison, and Andrew W. Fitzgibbon. Kinectfusion: real-time dynamic 3d surface reconstruction and interaction. In *SIGGRAPH*, 2011.

[23] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 1976.

[24] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *ICCV*, 2017.

[25] Junha Lee, Seungwook Kim, Minsu Cho, and Jaesik Park. Deep hough voting for robust global registration. In *ICCV*, 2021.

[26] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. *ICCV*, 2019.

[27] Lin Li, Wendong Ding, Yongkun Wen, Yufei Liang, Yong Liu, and Guowei Wan. A unified bev model for joint learning of 3d local features and overlap estimation, 2023.

[28] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A. Chapman, Dongpu Cao, and Jonathan Li. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[29] Tony Lindeberg. *Scale Invariant Feature Transform*, volume 7. 05 2012.

[30] Yinlong Liu, Chen Wang, Zhijian Song, and Manning Wang. Efficient global point cloud registration by matching rotation invariant features through translation search. In *ECCV*, 2018.

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[32] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-net: Towards learning based lidar localization for autonomous driving. In *CVPR*, 2019.

[33] Guofeng Mei, Xiaoshui Huang, Jian Zhang, and Qiang Wu. Overlap-guided coarse-to-fine correspondence prediction for point cloud registration. In *ICME*, 2022.

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018.

[35] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *CVPR*, 2020.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

[37] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[39] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, 2022.

[40] Szymon Rusinkiewicz. A symmetric objective function for icp. *ACM Transactions on Graphics (TOG)*, 2019.

[41] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, 2001.

[42] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009.

[43] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, 2009.

[44] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *(IROS)*, 2018.

[45] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 1967.

[46] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021.

[47] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019.

[48] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *ECCV*, 2010.

[49] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *arXiv preprint arXiv:2006.13566*, 2020.

[50] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *TPAMI*, 1991.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[52] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *ACM MM*, 2022.

[53] Tong Wei, Yash Patel, Alexander Shekhovtsov, Jiri Matas, and Daniel Barath. Generalized differentiable ransac, 2023.

[54] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015.

[55] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *ICCV*, 2021.

[56] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022.

[57] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *ECCV*, 2018.

[58] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *CVPR*, 2020.

[59] Zi Jian Yew and Gim hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *CVPR*, 2022.

[60] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018.

[61] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *NeurIPS*, 2021.

[62] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *ECCV*, 2020.

[63] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.