CoFiNet: Reliable Coarse-to-fine Correspondences for Robust Point Cloud Registration

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of extracting correspondences for point cloud registration. Existing works widely adopt keypoint detection before establishing correspondences but struggle to propose repeatable keypoints. To address this issue, we present CoFiNet - **Co**arse-to-**Fi**ne **Net**work which extracts hierarchical correspondences from coarse to fine without detecting keypoints. On coarse scale and guided by a weighting scheme, our model firstly learns to match down-sampled nodes whose vicinity points are overlapped and proposes node correspondences, which significantly shrinks the search space of a consecutive stage. On a finer scale, node proposals are consecutively expanded to patches that consist of groups of points together with associated descriptors. Patch correspondences are then refined to point level by a density-adaptive matching module capable to deal with varying point density. We extensively evaluate CoFiNet on both indoor and outdoor standard benchmarks. Results show that CoFiNet achieves state-of-the-art performance on all evaluated datasets. Especially on 3DLoMatch where point clouds share less overlap, CoFiNet significantly outperforms existing methods by at least 5% on Registration Recall, with at most two-third of parameters compared to them.

1 Introduction

Correspondence estimation is a core topic of computer vision and establishing reliable correspondences is a key to success in many fundamental vision tasks, such as tracking, reconstruction, flow estimation, and particularly, point cloud registration. Point cloud registration aims at recovering the transformation between a pair of partially overlapped point clouds. It is a fundamental task in a wide range of real applications, including scene reconstruction, autonomous driving, simultaneous localization and mapping (SLAM), etc. However, due to the unordered and irregular properties of point clouds, extracting reliable correspondences from them has been a challenging task for a long time. From early-stage hand-crafted methods [1, 2, 3, 4, 5] to recently emerged deep learning-based approaches [6, 7, 8, 9, 10, 11, 12, 13, 14, 15], many works contributed to improving the quality of correspondences.

Similar to other computer vision tasks, point cloud registration also benefits from the trends of deep neural networks. We can broadly categorize recent deep learning-based methods into two categories. The first [10, 11, 12] follows the idea of ICP [16], where they iteratively find dense correspondences and compute pose estimation. Though they can achieve reasonable results on synthetic object dataset [17], they struggle on large-scale scene benchmarks [18, 6]. This is mainly due to low correspondence precision and high computational complexity. Differently, the other category of methods [6, 7, 8, 9, 13, 15] tackles point cloud registration in a two-stage manner. They first learn local descriptors of sparser and more distinctive keypoints for matching, and afterwards use robust pose estimators, e.g., RANSAC [19], for recovering the relative transformation. Their two-stage

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

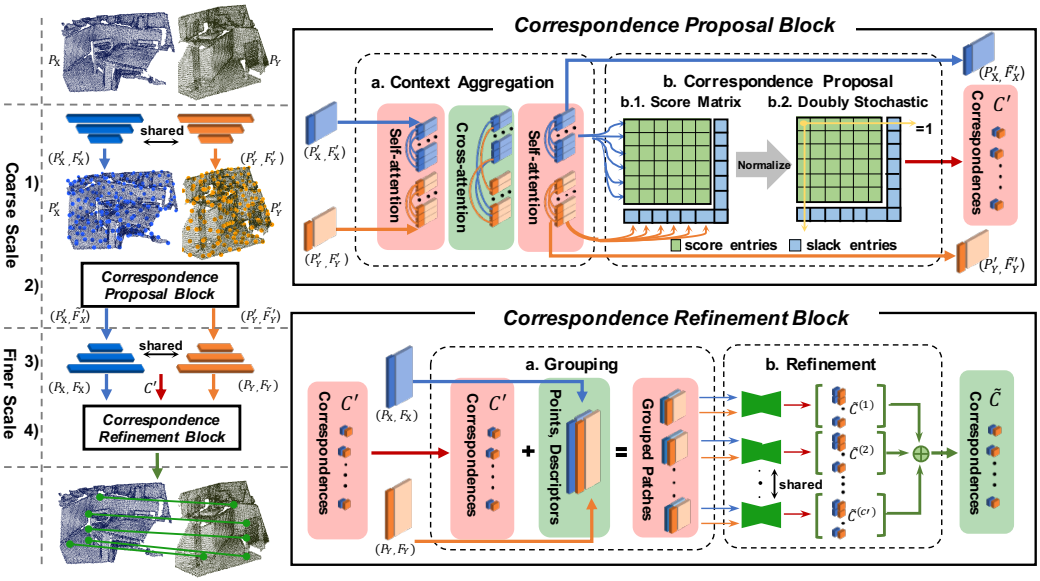

Figure 1: **Left**: Overview of our proposed CoFiNet. From top to bottom: 1) it first consumes a pair of point clouds and down-samples them to uniformly distributed nodes, while jointly learning their associated features; 2) features are strengthened and coarse node correspondences are proposed by the Correspondence Proposal Block; 3) strengthened features are decoded to dense descriptors associated to input points; 4) in the Correspondence Refinement Block, coarse node proposals are expanded to patch correspondences via grouping, which are then refined to point level by our proposed density-adaptive matching module in Fig. 2. **Top Right**: Illustration of the Correspondence Proposal Block. **Bottom Right**: Demonstration of the Correspondence Refinement Block.

strategy makes them achieve state-of-the-art performance on large-scale scene benchmarks [18, 6]. However, leveraging sparser keypoints either by uniform sampling [7, 8] or with salient point detection [20, 13, 15] brings side effects. The sparsity of keypoints by nature challenges their repeatability, i.e., it increases the risk that a certain keypoint loses its corresponding point after sampling. Recent approaches [9, 20, 13, 15] all adopt uniform sampling or salient point detection, which constrains their performance due to low keypoint repeatability. In 2D image matching tasks, which target establishing correspondences between a pair of images, recently emerged approaches [21, 22, 23] leverage a coarse-to-fine mechanism to avoid direct keypoint detection which shows superiority over the state-of-the-art detection-based methods [24]. However, in 3D point cloud matching, where keypoint detectors usually work worse than their counterparts in 2D, existing deep learning methods do not yet exploit such a coarse-to-fine strategy. To fill the gap, we focus on leveraging the coarse-to-fine mechanism to eliminate the side effects of sparse keypoints.

However, designing such a coarse-to-fine pipeline in point cloud matching is non-trivial, mainly due to the inherent unordered and irregular nature of point clouds. To this end, we propose CoFiNet that leverages a weighting scheme for coarse node matching and a density-adaptive matching module to deal with the varying density of points on finer scale, which enables the coarse-to-fine correspondence extraction from point clouds. For unordered point clouds, we stack KPConv [25] layers to sub-sample point clouds to uniformly distributed nodes, while to aggregate vicinity information simultaneously. Upon down-sampled nodes and learned features, we adopt a KPConv-based decoder to recover the raw resolution and to obtain dense descriptors. For irregularity, on coarse scale, we leverage a weighting scheme proportional to overlap ratios to guide training, which shows more robustness against varying density. On finer scale, we propose a density-adaptive matching module, which refines coarse correspondence proposals to point level by solving a differentiable optimal transport problem [24] with awareness to varying point density. Our main contributions are summarized as follows:

- A detection-free learning framework that treats point cloud registration as a coarse-to-fine correspondence problem, where point correspondences are consecutively refined from coarse proposals that are extracted from unordered and irregular point clouds.

- A weighting scheme that, on coarse scale, guides our model to learn to match uniformly down-sampled nodes whose vicinity points are overlapped, which significantly shrinks the search space for the refinement.
- A differentiable density-adaptive matching module that refines coarse correspondences to point level based on solving an optimal transport problem with awareness to point density, which is robust to scenarios where point density varies severely.

To the best of our knowledge, we are the first deep learning-based work that incorporates a coarse-to-fine mechanism in correspondence search for point cloud registration. A series of experiments are conducted on both indoor and outdoor benchmarks to show our superiority over other methods. Notably, CoFiNet surpasses other state-of-the-art methods with much fewer parameters. Specifically, compared to [15], we only use around two-third and one-fourth of parameters on indoor and outdoor benchmarks, respectively. We will make the code available soon.

2 Related Work

Learned 3D local descriptors Early attempts for learning 3D local descriptors are mainly based on local patches. As a pioneer, Zeng et al. [6] propose the 3DMatch benchmark, on which they exploit a Siamese network [26] that consumes voxel grids of TDFs (Truncated Distance Fields) to match local patches. To learn rotation-invariant features, Gojcic et al. [9] replace TDF with a voxelized smoothed density value (SDV) representation aligned to the local reference frame (LRF). PPFNet [7] directly consumes raw points augmented with point-pair features (PPF) by leveraging PointNet [27] as its encoder. To address the issue of the rotation-variant point coordinates, PPF-FoldNet [8] leverages only PPF, which is naturally rotation-invariant, as its input and further incorporates a FoldingNet [28] architecture to enable unsupervised training. To extract geometrical features, Graphite [14] utilizes graph neural networks for local patch description. SpinNet [29] utilizes LRF for alignment and 3D cylindrical convolution layers for feature extraction, achieving the best generalization ability across unseen datasets. However, patch-based methods suffer from severe problems of computational efficiency, as tpically shared activations of adjacent patches are not reused which introduces a large amount of redundant computation. To address this, FCGF [30] makes a first attempt by using sparse convolutions [31]. It computes dense descriptors of the whole input in a single pass leading to a 600x speed-up while being able to achieve comparable performance to patch-based methods. PREDATOR [15] first introduces an attention mechanism [32] for early information exchange in learning 3D local descriptors. Together with its keypoint sampling mechanism, PREDATOR achieves the state-of-the-art performance on both 3DMatch [6] and 3DLoMatch [15] benchmarks.

3D keypoint detectors Built upon uniformly down-sampled nodes, USIP [20] learns to regress the coordinates of the most salient point in vicinity of each down-sampled node in an unsupervised manner. However, it suffers from degenerated cases when the number of desired keypoints is relatively small. D3Feat [13] exploits a fully convolutional encoder-decoder architecture for joint dense detection and description. However, it does not take overlap relationships into consideration and shows poor performance in cases where the overlap areas of given point clouds are insufficient. In addition to jointly predicting dense salient scores and local descriptors, PREDATOR [15] also estimates dense overlap scores, which indicate the confidences of points laying on the overlap regions. Keypoints will be sampled under the condition of both saliency and overlap scores. Although it surpasses existing methods by a large margin on both 3DMatch and 3DLoMatch, it still suffers from the precision of predicted scores, especially that of salient scores, and the sparsity of sampled keypoints, which increases the ambiguity in matching.

Coarse-to-fine correspondences As witnessed in 2D image matching, many recent works [21, 22, 23] leverage a coarse-to-fine mechanism to eliminate the inherent repeatability problem in keypoint detection and thus boost performance. DRC-Net [21] utilizes 4D cost volumes to enumerate all the possible matches and establishes pixel correspondences in a coarse-to-fine manner. Concurrently with DRC-Net, Patch2Pix [22] first establishes patch correspondences and then regresses pixel correspondences according to matched patches. In a similar coarse-to-fine manner with Patch2Pixel, LoFTR [23] leverages Transformers [32], together with an optimal transport matching layer [24], to match mutual-nearest patches on the coarse level, and then refines the corresponding pixel of the patch center on the finer level.

3 Methodology

Problem statement Given a pair of unordered point sets $\mathbf{X}$ with n points $x_i \in \mathbb{R}^3$ and $\mathbf{Y}$ with m points $y_j \in \mathbb{R}^3$, we aim at recovering the rigid transformation $\overline{\mathbf{T}}^{\mathbf{X}}_{\mathbf{Y}} \in SE(3)$ between them. We define their corresponding coordinate matrices as $\mathbf{P}_{\mathbf{X}} \in \mathbb{R}^{n\times 3}$ and $\mathbf{P}_{\mathbf{Y}} \in \mathbb{R}^{m\times 3}$ for simpler notation. We follow the path of extracting correspondences first and then estimating the relative pose, where we mainly focus on the former. For this purpose, we propose CoFiNet that takes a pair of point clouds as input and outputs point correspondences, which can be leveraged to estimate the rigid transformation by RANSAC [19]. An overview of our proposed method can be found in Fig. 1.

3.1 Coarse-scale Matching

Point encoding On the coarse level, our target is to propose patch correspondences that consist of patches with sufficient overlaps. To achieve this goal, we first adopt shared KPConv [25] encoders to down-sample raw point clouds to nodes $\mathbf{P}'_{\mathbf{X}} \in \mathbb{R}^{n'\times 3}$ and $\mathbf{P}'_{\mathbf{Y}} \in \mathbb{R}^{m'\times 3}$, while jointly learning their associated features $\mathbf{F}'_{\mathbf{X}} \in \mathbb{R}^{n'\times b}$ and $\mathbf{F}'_{\mathbf{Y}} \in \mathbb{R}^{m'\times b}$. Demonstration of down-sampled nodes can be found in the middle of the left part in Fig. 1.

Attentional feature aggregation As illustrated in Fig. 1, the Correspondence Proposal Block (CPB) takes as input the down-sampled nodes and associated features. Before using features $\mathbf{F}'_{\mathbf{X}}$ and $\mathbf{F}'_{\mathbf{Y}}$ to calculate the score matrix, we do an extra step following [24, 15], where the attention [32] is leveraged to strengthen the distinctiveness of features. Following [15], we adopt a sequence of self-, cross- and self-attention, which interactively incorporates global contexts across nodes from the same point cloud and the opposite point cloud. Below we briefly introduce the cross-attention module. Given $\mathbf{F}'_{\mathbf{X}}$ and $\mathbf{F}'_{\mathbf{Y}}$, akin to database retrieval, the former is linearly projected by a learnable matrix $\mathbf{W}_{\mathbf{Q}} \in \mathbb{R}^{b\times b}$ to *query* $\mathbf{Q}$ as $\mathbf{Q} = \mathbf{F}'_{\mathbf{X}}\mathbf{W}_{\mathbf{Q}}$, while the latter is similarly projected to *key* $\mathbf{K}$ and *value* $\mathbf{V}$ by learnable matrices $\mathbf{W}_{\mathbf{K}}$ and $\mathbf{W}_{\mathbf{V}}$, respectively. The attention matrix $\mathbf{A}$ is represented as $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top/\sqrt{b}$, whose rows are then normalized by *softmax*. The message $\mathbf{M}$ flows from $\mathbf{F}'_{\mathbf{Y}}$ to $\mathbf{F}'_{\mathbf{X}}$ is formulated as $\mathbf{M} = \mathbf{A}\cdot\mathbf{V}$, which represents the linear combination of *values* weighted by the attention matrix. In the cross-attention module, contexts are aggregated in both directions, from $\mathbf{F}'_{\mathbf{X}}$ to $\mathbf{F}'_{\mathbf{Y}}$ and from $\mathbf{F}'_{\mathbf{Y}}$ to $\mathbf{F}'_{\mathbf{X}}$. For the self-attention module, we replace the graph-based module [33] leveraged in [15] with the cross attention module, but with a pair of identical features as input, i.e., information will be aggregated across nodes from the same point cloud.

Correspondence proposal As shown in Fig. 1, in the CPB and after the attention modules, we leverage strengthened features $\tilde{\mathbf{F}}'_{\mathbf{X}}$ and $\tilde{\mathbf{F}}'_{\mathbf{Y}}$ to calculate the score matrix. We match down-sampled nodes according to the overlap ratios of their vicinity points. There can be two cases where a node fails to match: 1) the major portion of its vicinities is occluded in another frame; and 2) its vicinities are visible, but no node in another frame has sufficient overlaps with it. Thus, for the score matrix, we expand it with a slack row and column that have m' and n' slack entries, respectively [34]. Nodes that fail to match other nodes would match to corresponding slack entries, i.e., have maximum scores there. Similar to [24], we compute the score matrix using inner product, which can be presented as:

$$\mathbf{S}' = \begin{bmatrix} \tilde{\mathbf{F}}'_{\mathbf{X}}\tilde{\mathbf{F}}'^{\top}_{\mathbf{Y}} & \mathbf{z} \\ \mathbf{z} & z \end{bmatrix}, \qquad \mathbf{S}' \in \mathbb{R}^{(n'+1)\times(m'+1)}, \tag{1}$$

where all slack entries are set to the same learnable parameter z. On $\mathbf{S}'$we run the Sinkhorn Algorithm [35, 36, 37] to make it doubly stochastic and represent correspondence probabilities between node of row i' and node of column j'. To guarantee higher recall, instead of picking the maximum score in every row and column, we adopt a threshold τ_c for likely correspondences whose scores are above τ_c. We define the obtained coarse node correspondence set as $\mathbf{C}' = \{(\mathbf{P}'_{\mathbf{X}}(i'), \mathbf{P}'_{\mathbf{Y}}(j'))\}$, with $|\mathbf{C}'| = c'$, where $|\cdot|$ denotes the set cardinality.

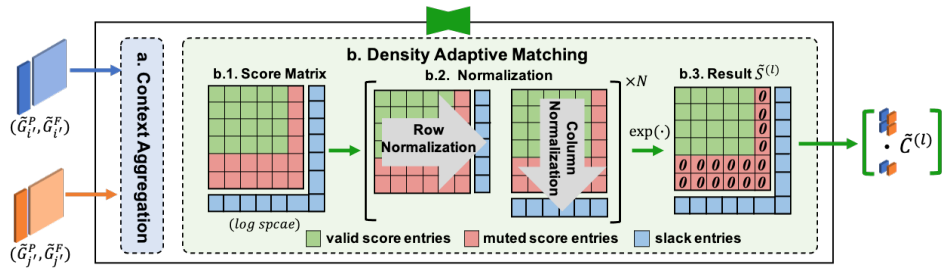

Figure 2: Illustration of our proposed density-adaptive matching module. The input is a pair of patches truncated by k. a) We use the context aggregation part from the Correspondence Proposal Block to condition on both patches and to strengthen features. b.1) The score matrix is computed. Slack entries are initialized with 0 and muted score entries corresponding to repeated sampled points are set to $-\infty$. b.2) N iterations of the Sinkhorn Algorithm are performed. We drop the slack row and column for row and column normalization, respectively. b.3) We obtain the final score matrix, whose first k rows and k columns are row- and column-normalized, respectively. For correspondences, we pick the maximum score in every row and column to guarantee higher precision.

3.2 Point-level Refinement

Node decoding On the finer level, we aim at refining coarse correspondences from the preceding stage to point level, which can then be used for point cloud registration. First, we stack several KPConv [38] layers to recover the raw points, $\mathbf{P_X}$ and $\mathbf{P_Y}$, while jointly learning associated dense descriptors, $\mathbf{F_X} \in \mathbb{R}^{n \times c}$ and $\mathbf{F_Y} \in \mathbb{R}^{m \times c}$. We thereby assign to each point p an associated feature $p \leftrightarrow f \in \mathbb{R}^c$. As demonstrated in the bottom right of Fig. 1, obtained dense descriptors, together with raw points and coarse correspondences are fed into the Correspondence Refinement Block (CRB), where coarse proposals are expanded to patches which are then refined to point correspondences.

Point-to-node grouping For refinement, we need to expand nodes in coarse correspondences to patches, i.e., groups of points and their descriptors. For doing this, we use a point-to-node grouping strategy [38, 39, 20] to associate points to their nearest nodes. The advantages of point-to-node over k-nearest neighbor search or radius-based ball query are two-fold: 1) Every point will be assigned to a node, while some points could be left out in other strategies. 2) It can automatically adapt to various scales [20]. After grouping, nodes with their associated points and descriptors form patches, upon which we can extract point correspondences. For a certain node $\mathbf{P'_X}(i')$, its associated point set $\mathbf{G^P_{i'}}$ and feature set $\mathbf{G^F_{i'}}$ can be denoted as:

$$\begin{cases} \mathbf{G^P_{i'}} = \{p \in \mathbf{P_X} \big| \|p - \mathbf{P'_X}(i')\| \leq \|p - \mathbf{P'_X}(j')\| \; \forall \, i' \neq j'\}, \\ \mathbf{G^F_{i'}} = \{f \in \mathbf{F_X} \big| f \leftrightarrow p \text{ with } p \in \mathbf{G^P_{i'}}\}, \end{cases} \tag{2}$$

where $\|\cdot\| = \|\cdot\|_2$ represents the Euclidean norm. In the point-to-node grouping, we expand the coarse node correspondence set $\mathbf{C'}$ to coarse patch correspondence sets, both in geometry space $\mathbf{C_P} = \{(\mathbf{G^P_{i'}}, \mathbf{G^P_{j'}})\}$ and feature space $\mathbf{C_F} = \{(\mathbf{G^F_{i'}}, \mathbf{G^F_{j'}})\}$.

Density-adaptive matching Extracting point correspondences from a pair of roughly matched patches is the same as matching two partially overlapped point clouds, but with a smaller scale and from a local perspective. Thus, directly leveraging the CPB in Fig. 1 with input $(\mathbf{G^P_{i'}}, \mathbf{G^F_{i'}})$ and $(\mathbf{G^P_{j'}}, \mathbf{G^F_{j'}})$ could theoretically tackle the problem. However, simply utilizing CPB to extract point correspondences would lead to a bias towards slack rows and columns, i.e., the model would tend to predict every point as occluded. Reasons for this are two-fold: 1) For computational efficiency, similar to radius-based ball query, in a point-to-node grouping, we need to truncate the number of points to a unified number k for every patch. If a patch contains less than k points, like in [40], a fixed point or randomly sampled points will be repeated as a supplement. 2) On a coarse level, our model learns to match nodes whose vicinities have higher overlap ratios. However, due to the varying density of points, after expansion, patches could include only a few points. As a result, many occluded points would be repeated, which introduces the aforementioned bias during training. To address the issue, we propose a density-adaptive matching module that refines coarse correspondences to point level

by solving an optimal transport problem with awareness to point density. We denote the truncated patches as $\widetilde{\mathbf{G}}^{\cdot}_{\cdot}$ and demonstrate our proposed density-adaptive matching module in Fig. 2. Notably, both during and after normalization, the exponent projection of any muted score entries is always equal to 0, which eliminates the side effects caused by the repeated sampling of points. The final point correspondence set $\widetilde{\mathbf{C}}$ is represented as the union of all the obtained correspondence sets $\widetilde{\mathbf{C}}^{(l)}$, which can be directly leveraged by RANSAC[19] for registration.

3.3 Loss Functions

Our total loss $\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_f$ is calculated as the weighted sum of the coarse-scale $\mathcal{L}_c$ and the fine-scale $\mathcal{L}_f$ where λ is used to balance the terms. We detail the individual parts hereafter.

Coarse scale On coarse scale, we leverage a weighting scheme proportional to the overlap ratios over patches as coarse supervision. For this, we analyse the correspondences within patches. Two points p and q are hereby considered in correspondence $p \leftrightarrow q$ if their distance $\|\overline{\mathbf{T}}^{\mathbf{X}}_{\mathbf{Y}}(\mathbf{p}) - \mathbf{q}\|_2$ is below a certain threshold $\tau_p \in \mathbb{R}$. For two untruncated coarse patches $\mathbf{G}^{\mathbf{P}}_{i'}$ and $\mathbf{G}^{\mathbf{P}}_{j'}$ we calculate the average ratio of corresponding points which we use as a weight $\mathbf{W}'(i', j')$ to weight patch correspondences in a given score matrix $\mathbf{S}'$. Additionally, we define the values of slack entries as the ratio of points that belong to their corresponding patches but fail to match. Specially, the right bottom entry equals 0. With this, we define the coarse scale loss as:

$$\mathcal{L}_c = \frac{-\sum_{i',j'} \mathbf{W}'(i', j') \log(\mathbf{S}'(i', j'))}{\sum_{i',j'} \mathbf{W}'(i', j')}. \tag{3}$$

Finer scale On a finer scale, we define the ground truth correspondence matrix with slacks as a binary matrix $\widetilde{\mathbf{B}}^{(l)} \in \{0, 1\}^{(k+1)\times(k+1)}$ for the lth truncated patch correspondence where a value $\mathbf{B}^{(l)}(i, j) \in \{0, 1\}$ indicates whether a correspondence $i \leftrightarrow j$ exists or not. Moreover, 1 on slack entries means their corresponding points fail to match and 0 otherwise. Specially, the right bottom entry equals to 0. Therefore, the loss function on fine scale reads as:

$$\mathcal{L}_f = \frac{-\sum_{l,i,j} \widetilde{\mathbf{B}}^{(l)}(i, j) \log(\widetilde{\mathbf{S}}^{(l)}(i, j))}{\sum_{l,i,j} \widetilde{\mathbf{B}}^{(l)}(i, j)}, \tag{4}$$

where we define $0 \cdot \log(0) = 0$.

4 Results

We evaluate our model on three challenging public benchmarks, including both indoor and outdoor scenarios. Following [15], for indoor scenes, we evaluate our model on both 3DMatch [6], where point cloud pairs share > 30% overlaps, and 3DLoMatch [15], where overlaps of point cloud pairs are between 10% to 30%. In line with existing works [13, 15], we evaluate for outdoor scenes on odometryKITTI [18], where varying density of points is the major challenge.

4.1 Implementation details

CoFiNet is implemented in PyTorch [41] and can be trained end-to-end on a single RTX 2080Ti GPU. We train 25(60) epochs on 3DMatch/3DLoMatch (KITTI), with $\lambda = 1$ ($\lambda = 3$), both using Adam optimizer with an initial learning rate of 3e-4, which is exponentially decayed by 0.05 after each epoch. We adopt similar encoder and decoder architectures as [15], but with significantly fewer parameters. We use a batch size of 1 in all experiments. For training the attention-based network on finer scale, we sample 128 coarse correspondences, with truncated patch size $k = 64$ on 3DMatch (3DLoMatch). On KITTI, the numbers are 128 and 32, respectively. Specially, due to the severely-varying point density on KITTI, we use the weighting matrix $\mathbf{W}'$ for sampling to improve the overlap ratios of coarse correspondences, while binarizing it in loss functions. At test time, all the extracted coarse correspondences are fed into the finer stage for refinement, with the same k as in training. We use our proposed point correspondences and RANSAC [19] for point cloud registration.

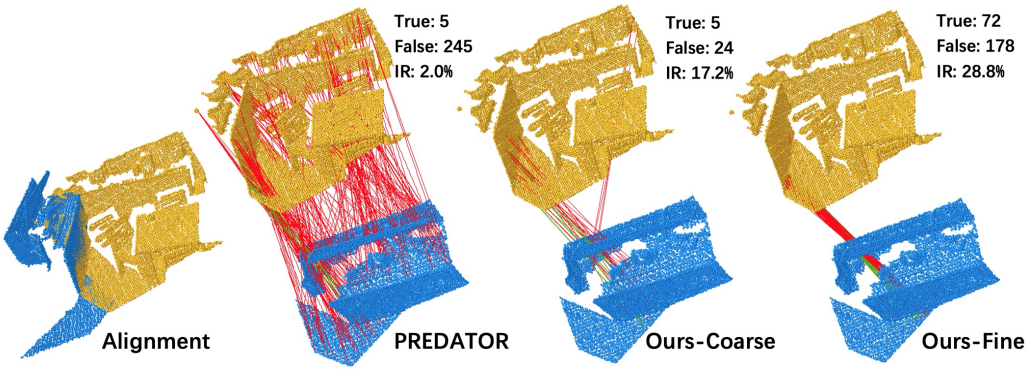

Figure 3: Qualitative results on *Inlier Ratio*. We compare our point correspondences (the last column) with our coarse correspondences (the third raw) and correspondences from PREDATOR (the second raw) on an example case from 3DLoMatch. The first column provides the ground truth alignment, which shows overlaps are very limited. The significantly larger inlier ratio can be observed from the incorrect (red) and correct (green) correspondence connections.

4.2 3DMatch and 3DLoMatch

Dataset 3DMatch [6] consists of 62 indoor scenes, where 46 scenes are used for training, 8 scenes for validation and 8 scenes for testing. We use the training set utilized in [15], which is a mixture of point cloud pairs with overlaps both larger than 30%, and between 10% and 30%. For testing, we evaluate our model on both 3DMatch and 3DLoMatch following the same evaluation protocol as [15].

Metrics We adopt three typically-used metrics, namely *Registration Recall* (RR), *Feature Matching Recall* (FMR) and *Inlier Ratio* (IR), to show the superiority of CoFiNet over existing approaches. Specifically, 1) the *Registration Recall* is the fraction of point cloud pairs whose error of transformation estimated by RANSAC is smaller than a certain threshold, e.g., RMSE < 0.2m, compared to the ground truth. 2) The *Feature Matching Recall* indicates the percentage of point cloud pairs whose *Inlier Ratio* is larger than a certain threshold, e.g., $\tau_2 = 5\%$. 3) The *Inlier Ratio* is the fraction of correspondences whose residual error in geometry space is less than a threshold, e.g., $\tau_1 = 10$cm, under the ground truth transformation.

Correspondence sampling We follow [13, 15] and report performance with different numbers of samples. However, as CoFiNet avoids keypoint detection and directly outputs point correspondences, we cannot strictly follow [13, 15] to sample different numbers of interest points. For fair comparison, we instead sample correspondences in our experiments, but keep the same number as them. We adopt two strategies for sampling a fixed number of point correspondences from our proposed correspondence set $\widetilde{\mathbf{C}}$, namely, *rand*, where each correspondence has equal possibility, and *prob*, where correspondences are sampled with probability proportional to a global confidence $c_{global} = c_{fine} \cdot c_{coarse}$. For a certain point correspondence, we define c_{fine} as its fine-level score and c_{coarse} as $\mathbf{S}'(i', j')$ s.t. the correspondence is refined from the truncated patch correspondence $(\widetilde{\mathbf{G}}^{\mathbf{P}}_{i'}, \widetilde{\mathbf{G}}^{\mathbf{P}}_{j'})$.

Inlier ratio As the main contribution of CoFiNet is that we adopt the coarse-to-fine mechanism to avoid keypoint detection, where existing methods struggle to detect repeatable keypoints, we first check the *Inlier Ratio* of CoFiNet, which is directly related with the keypoint repeatability. We show quantitative results in Tab. 1 and qualitative results in Fig. 3. As shown in Tab. 1, on *Inlier Ratio*, CoFiNet(*rand*) outperforms all previous methods on 3DLoMath and only performs slightly worse than PREDATOR [15] and FCGF[30] on 3DMatch. After we change the sampling strategy to *prob*, we obtain higher *Inlier Ratio* on both datasets and surpass PREDATOR[15] on 3DMatch, where we are only worse than FCGF[30] when sampled points are sufficient. Notably, as our method directly proposes correspondences, it is more robust when the number of samples is relatively small. However, for other detection-based methods, as they face a more severe repeatability problem when only few points are sampled, their performance drops rapidly. The fact that *prob* surpasses *rand* by a large margin indicates additionally that our model learns a good distribution of matching scores in both coarse and fine scale, i.e., higher confidences indicate a higher quality of correspondences.

Table 1: Results on both 3DMatch and 3DLoMatch datasets under different numbers of samples. We also show the number of utilized parameters of all the approaches in the last column.

	3DMatch					3DLoMatch					
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250	# Params ↓
Registration Recall(%) ↑											
3DSN[9]	78.4	76.2	71.4	67.6	50.8	33.0	29.0	23.3	17.0	11.0	10.24M
FCGF[30]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8	8.76M
D3Feat[13]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1	27.28M
PREDATOR[15]	88.3	88.3	**89.0**	**88.4**	84.7	54.2	55.8	56.7	56.1	50.7	7.43M
CoFiNet(*rand*)	**88.8**	**88.4**	87.8	87.1	**87.4**	62.1	61.5	**62.6**	**61.5**	58.0	**5.48M**
CoFiNet(*prob*)	87.4	88.1	86.6	86.4	85.9	**63.6**	**62.3**	61.8	60.3	**60.2**	**5.48M**
Feature Matching Recall(%) ↑											
3DSN[9]	95.0	94.3	92.9	90.1	82.9	63.6	61.7	53.6	45.2	34.2	10.24M
FCGF[30]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3	8.76M
D3Feat[13]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5	27.28M
PREDATOR[15]	96.6	96.2	96.2	95.8	95.5	71.7	73.8	73.8	72.9	72.3	7.43M
CoFiNet(*rand*)	97.7	**97.9**	**97.8**	**97.7**	**98.0**	81.6	81.6	**82.2**	81.3	81.3	**5.48M**
CoFiNet(*prob*)	**98.0**	97.8	**97.8**	97.6	97.7	**81.7**	**82.1**	82.0	**81.7**	**81.9**	**5.48M**
Inlier Ratio(%) ↑											
3DSN[9]	36.0	32.5	26.4	21.5	16.4	11.4	10.1	8.0	6.4	4.8	10.24M
FCGF[30]	**56.8**	**54.1**	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6	8.76M
D3Feat[13]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0	27.28M
PREDATOR[15]	49.9	50.3	49.2	46.3	41.8	20.0	20.8	21.0	20.2	19.0	7.43M
CoFiNet(*rand*)	45.1	45.1	45.1	45.1	45.0	26.9	26.9	26.9	26.8	26.8	**5.48M**
CoFiNet(*prob*)	50.4	51.1	**51.6**	**51.7**	**52.0**	**28.9**	**29.8**	**30.4**	**30.6**	**30.8**	**5.48M**

Comparison to state-of-the-art approaches We compare our proposed CoFiNet to other state-of-the-art approaches including 3DSN [9], FCGF [30], D3Feat [13] and PREDATOR [15] in Tab. 1 and Fig. 4. For *Feature Matching Recall*, both CoFiNet(*rand*) and CoFiNet(*prob*) outperform all other methods on 3DMatch and 3DLoMatch. Especially on 3DLoMatch, which is more challenging due to the low-overlap scenario, our proposed method surpasses others with a large margin of more than 8%. It indicates that CoFiNet is more robust to different scenes, i.e., we find at least 5% true correspondences for more test cases. Additionally, we also follow [13, 15] to show the *Feature Matching Recall* in relation to τ_2 and τ_1 on 3DMatch, which further show our supiriority over other methods. When referring to the most important metric *Registration Recall* that better reflects the final performance on point cloud registration, though we report comparable performance to PRDATOR [15], we significantly outperform others on 3DMatch. When evaluated on 3DLoMatch, our proposed approach significantly surpasses others by more than 5%, which shows the advantages of our method in scenarios with less overlaps. Besides that, our method is more robust when only few points can be sampled as indicated by the *Inlier Ratio*. Moreover, we also compare the number of parameters used in different methods in the last column of Tab.1, which shows that CoFiNet uses the least parameters while achieving the best performance.

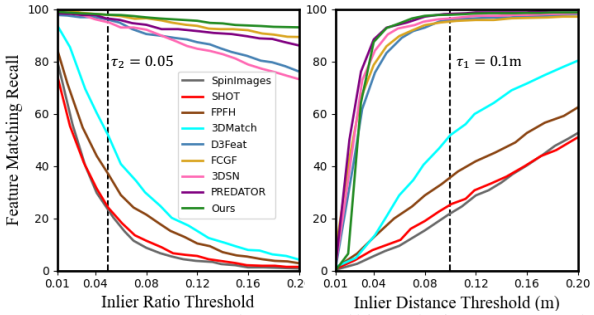

Figure 4: *Feature Matching Recall* in relation to: 1) *Inlier Ratio Threshold* (τ_2) and 2) *Inlier Distance Threshold* (τ_1) on 3DMatch.

Ablation study To get a better understanding of CoFiNet, we ablate different modules of CoFiNet. Related results can be found in Tab. 2. In the first experiment, we directly use the coarse correspondence set $\mathbf{C}'$ for point cloud registration. Unsurprisingly, it performs worse on all the three metrics, compared to our all-combined model, which indicates CoFiNet benefits from refinement. Then, we ablate the weighting scheme proportional to overlap ratios, which guides the coarse matching of patches. We replace it with a binary mask similar to the one used on the fine level. Results

show that it leads to a worse performance, which indicates that coarse matching of patches benefits from our designed weighting scheme. Finally, we do the last ablation study on the density-adaptive matching module. Results indicate that on both 3DMatch and 3DLoMatch, without the density-adaptive matching module, CoFiNet can still achieve on-par performance with our all-combined model. This is because point cloud density varies only insignificantly for indoor scenarios. Notably, the density-adaptive matching module enables CoFiNet to work on KITTI, where point density varies severely.

Table 2: Ablation study of our proposed CoFiNet on 3DMatch, 3DLoMatch and KITTI. We set # samples = 2500 for indoor datasets and use the *prob* strategy for all tests. Best performance is highlighted in bold.

	3DMatch			3DLoMatch			KITTI		
	RR (%)↑	FMR (%)↑	IR (%) ↑	RR (%)↑	FMR (%)↑	IR↑	RTE (cm)↓	RRE (°)↓	RR (%)↑
Full CoFiNet pipeline	**88.1**	**97.8**	51.1	**62.3**	**82.1**	**29.8**	**8.2**	**0.41**	**99.8**
w/o refinement	79.4	96.0	42.4	40.3	81.0	24.5	47.8	1.75	79.8
w/o weighting	86.4	97.5	46.8	59.7	81.6	**29.8**	12.9	0.58	98.4
w/o density-adaptive	88.0	**97.8**	**52.4**	62.2	81.4	27.8	failed	failed	failed

4.3 KITTI

Dataset KITTI [18] includes 11 sequences of outdoor driving scenarios, which are scanned with a LiDAR sensor. We follow [15] for data processing and training. The major challenge of KITTI comes from its varying point density. We evaluate our model on KITTI to prove its capability to deal varying density.

Metrics We follow [15] and use 3 metrics, namely, the *Relative Rotation Error* (RRE), which is the geodesic distance between estimated and ground truth rotation matrices, the *Relative Translation Error* (RTE), which is the Euclidean distance between the estimated and ground truth translation, and the *Registration Recall* (RR) mentioned before.

Table 3: Quantitative comparisons on KITTI.

Method	RTE(cm)↓	RRE(°)↓	RR(%)↑	Params↓
3DFeat-Net[42]	25.9	0.57	96.0	**0.32M**
FCGF[30]	9.5	0.30	96.6	8.76M
D3Feat[13]	7.2	0.30	**99.8**	27.28M
PREDATOR[15]	**6.8**	**0.27**	**99.8**	22.77M
CoFiNet(*rand*)	8.6	0.41	**99.8**	5.46M
CoFiNet(*prob*)	8.2	0.41	**99.8**	5.46M

Comparison to state-of-the-art approaches On KITTI, we compare CoFiNet to 3DFeat-net [42], FCGF [30], D3Feat [13] and PREDATOR [15]. Quantitative results can be found in Tab. 3. On RTE and RRE, we stay in the middle, but for RR, together with [13, 15], we perform best. Notably, we achieve such performance by using only 5.46M parameters and a training for 60 epochs compared to the best performing model [15], which uses over 20M parameters and is trained for 150 epochs. This experiment indicates that our model is able to deal with outdoor scenarios where point density varies severely.

5 Conclusion

In this paper, we present a deep neural network that leverages a coarse-to-fine strategy to extract correspondences from unordered and irregularly sampled point clouds for registration. Our proposed model is capable of directly consuming unordered point sets and proposing reliable correspondences without the assistance of keypoints. To tackle the irregularity of point clouds, on coarse scale, we first propose a weighting scheme proportional to overlap ratios of patches, which guides the model to match nodes whose vicinity points are overlapped and significantly shrinks the search space of a following refinement. On finer level, we then adopt a density-adaptive matching module which eliminates the side effects from repeated sampling and enables our model to deal with density varying points. Extensive experiments on both indoor and outdoor benchmarks validate the effectiveness of our proposed model. We stay on-par with state-of-the-art approaches on 3DMatch and KITTI, while significantly surpassing previous approaches on 3DLoMatch using a model with significantly fewer parameters.

References

[1] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[2] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[3] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique shape context for 3d data description. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 57–62, 2010.

[4] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[5] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Persistent point feature histograms for 3d point clouds. In *Proc 10th Int Conf Intel Autonomous Syst (IAS-10), Baden-Baden, Germany*, pages 119–128, 2008.

[6] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017.

[7] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 195–205, 2018.

[8] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018.

[9] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019.

[10] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019.

[11] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *arXiv preprint arXiv:1910.12240*, 2019.

[12] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020.

[13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6359–6367, 2020.

[14] Mahdi Saleh, Shervin Dehghani, Benjamin Busam, Nassir Navab, and Federico Tombari. Graphite: Graph-induced feature extraction for point cloud registration. In *2020 International Conference on 3D Vision (3DV)*, pages 241–251, 2020.

[15] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. *arXiv preprint arXiv:2011.13005*, 2020.

[16] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[17] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[19] 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*, 24(6):381–395, 1981.

[20] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 361–370, 2019.

[21] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[22] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. *arXiv preprint arXiv:2012.01909*, 2020.

[23] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *arXiv preprint arXiv:2104.00680*, 2021.

[24] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[25] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019.

[26] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6:737–744, 1993.

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[28] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018.

[29] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. *arXiv preprint arXiv:2011.12149*, 2020.

[30] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019.

[31] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

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

[34] Benjamin Busam, Marco Esposito, Simon Che'Rose, Nassir Navab, and Benjamin Frisch. A stereo vision approach for cooperative robotic movement therapy. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 127–135, 2015.

[35] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[36] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

[37] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[38] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[39] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018.

[40] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[41] 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. *arXiv preprint arXiv:1912.01703*, 2019.

[42] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 607–623, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contribution and scope? [Yes] The main claims are presented in Section 1 and they are supported by experiments in Section 4.
   (b) Have you read the ethics review guidelines and ensured that your paper confirms to them? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please refer to the discussion section of supplemental material.
   (d) Did you describe the limitations of your work? [Yes] Please refer to the discussion section of supplemental material.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Code and data will be released publicly.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please refer to Section 4.1, 4.2 and 4.3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please refer to Section 4.1 and the "Params" columns of Table 1 and 3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We use indoor public benchmarks 3DMatch[6] and 3DLoMatch[15] as well as the outdoor autonomous driving data odometryKITTI[18].

(b) Did you mention the license of the assets? [Yes] Please refer to the data section of supplemental material.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Please refer to the data section of supplemental material.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Please refer to the data section of supplemental material.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]