BUFFER-X: Towards Zero-Shot Point Cloud Registration in Diverse Scenes

Minkyun Seo[1*] Hyungtae Lim[3*] Kanghee Lee[1] Luca Carlone[3] Jaesik Park[1,2†]
[1]Computer Science Engineering and [2]Interdisciplinary Program of AI, Seoul National University
[3]Laboratory for Information & Decision Systems, Massachusetts Institute of Technology
minkyunseo00@gmail.com, shapelim@mit.edu, kanghee.lee@snu.ac.kr,
lcarlone@mit.edu, jaesik.park@snu.ac.kr

Abstract

Recent advances in deep learning-based point cloud registration have improved generalization, yet most methods still require retraining or manual parameter tuning for each new environment. In this paper, we identify three key factors limiting generalization: (a) reliance on environment-specific voxel size and search radius, (b) poor out-of-domain robustness of learning-based keypoint detectors, and (c) raw coordinate usage, which exacerbates scale discrepancies. To address these issues, we present a zero-shot registration pipeline called BUFFER-X by (a) adaptively determining voxel size/search radii, (b) using farthest point sampling to bypass learned detectors, and (c) leveraging patch-wise scale normalization for consistent coordinate bounds. In particular, we present a multi-scale patch-based descriptor generation and a hierarchical inlier search across scales to improve robustness in diverse scenes. We also propose a novel generalizability benchmark using 11 datasets that cover various indoor/outdoor scenarios and sensor modalities, demonstrating that BUFFER-X achieves substantial generalization without prior information or manual parameter tuning for the test datasets. Our code is available at https://github.com/MIT-SPARK/BUFFER-X.

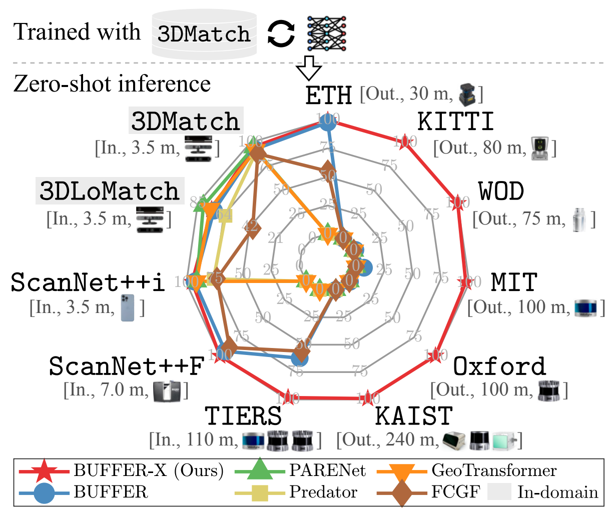


Fig. 1: Sucess rate (unit: %) of *zero-shot point cloud registration* with state-of-the-art approaches on 11 datasets [27, 31, 33, 55, 59, 60, 68, 70, 80, 87]. Without any prior information or manual parameter tuning for the test datasets, our *BUFFER-X* shows robust generalization capability across diverse scenes even though the network is only trained on the 3DMatch dataset [87].

1. Introduction

The field of deep learning-based point cloud registration has made steady and remarkable progress, including enhancing feature distinctiveness [2, 4–6, 14, 15, 31, 48], improving data association strategies [25, 31, 43–45, 86, 89, 90], and developing more robust pose estimation solvers [32, 66, 78, 93]. Consequently, existing approaches achieve strong performance on test sequences within the same dataset used for training, successfully estimating the relative pose between two partially overlapping point clouds [10, 39, 42, 76, 77, 83].

More recently, there has been growing interest in tackling the generalization of these deep learning-based methods [5, 6, 16, 24, 53], which is the capability of a network to perform well across diverse real-world scenarios.

While these approaches have demonstrated excellent generalization performance, in practice, most existing methods still require the user to provide optimal parameters, such as voxel size for downsampling cloud points and search radius for descriptor generation, when dealing with unseen domain datasets. In this paper, we refer to this manual tuning as an *oracle*. Therefore, it is still desirable to develop zero-shot registration approaches for better usability and practical deployment.

[*]These authors contributed equally to this work.
[†]Corresponding author.

Furthermore, despite nearly a decade of research in deep learning-based registration, most studies remain confined to specific scenarios, primarily conducting experiments using omnidirectional LiDAR point clouds for outdoor environments [27, 81] and RGB-D depth clouds for indoor settings [87]. For this reason, domain generalization experiments on LiDAR point clouds in indoors [59] or with different LiDAR scan patterns in outdoor environments [33, 38, 59] are less explored. This underscores the need for a new benchmark that better reflects real-world sensor variations to evaluate generalizability across unseen environments and diverse scanning patterns.

In this context, the main contribution of this paper is addressing two key issues above and propose: a) a *zero-shot registration* architecture and b) a novel benchmark to help evaluate the generalization capability of deep learning-based registration approaches, as shown in Fig. 1. First, inspired by the remarkable generalization of BUFFER [5] as long as a user manually tunes the voxel size and search radius, we first thoroughly analyze the architectural principles that underpin its generalization. Then, we identify three factors that hinder the zero-shot capabilities of existing methods in Sec. 3. Building on these insights, we introduce a self-adaptive mechanism to determine the optimal voxel size for each test scene and streamline the pipeline of BUFFER, ultimately presenting a robust multi-scale patch-wise approach. We name our approach *BUFFER-X* to signify that it is an extension of BUFFER.

Second, we establish a comprehensive benchmark that encompasses both indoor and outdoor settings, ensuring that outdoor settings include culturally and geographically diverse locations (*e.g.,* captured in Europe, Asia, and the USA), various environmental scales (ranging from meters to kilometers), and different LiDAR scanning patterns, while indoor settings also incorporate LiDAR-captured data. Subsequently, we demonstrate that our method achieves promising generalization capability without any prior information or manual parameter tuning during the evaluation; see Fig. 1.

In summary, we make three key claims: (i) we thoroughly analyze the limitations of existing approaches and identify the key factors that have hindered zero-shot generalization, (ii) we present an improved approach, named BUFFER-X, that addresses the generalization issues of state-of-the-art methods, and (iii) we introduce a benchmark to evaluate zero-shot generalization performance comprehensively.

2. Related Work

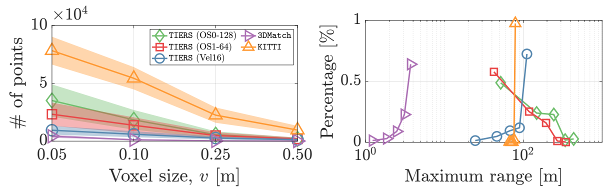
3D point cloud registration, which estimates the relative pose between two partially overlapping point clouds, is a fundamental problem in the fields of robotics and computer vision [7, 13, 17, 36, 82]. Overall, point cloud registration methods are classified into two categories based on whether their performance relies on the availability of an initial guess for registration: a) *local* registration [11,35,54,56,65, 71] and b) *global* registration [10, 23, 26, 39, 42, 75–77, 93]. Global registration methods can be further classified into two types: a) *correspondence-free* [10, 12, 18, 22, 30, 49–51, 61, 64, 92] and b) *correspondence-based* [23, 26, 37, 43, 77, 85, 86, 93] approaches. In this study, we focus on the latter and particularly place more emphasis on deep learning-based registration methods.

Since Qi *et al.* [57] demonstrated that learning-based techniques in 2D images can also be applied to 3D point clouds, a wide range of learning-based point cloud registration approaches have been proposed. Building on these advances, novel network architectures with increased capacity have continuously emerged, ranging from MinkUNet [19–21], cylindrical convolutional network [5,6,94], KPConv [8, 31, 69] to Transformers [15, 58, 72, 73].

While these advancements have led to improved registration performance, some of these methods often exhibit limited generalization capability, leading to performance degradation when applied to point clouds captured by different sensor configurations or in unseen environments. To tackle the generalization problem, Ao *et al.* [3, 6] introduce SpinNet, a patch-based method that normalizes the range of local point coordinates within a fixed-radius neighborhood to $[-1, 1]$. This makes us come to realize that patch-wise scale normalization is key to achieving a data-agnostic registration pipeline.

Further, Ao *et al.* [5] proposed BUFFER to enhance efficiency by combining point-wise feature extraction with patch-wise descriptor generation. However, we found that such learning-based keypoint detectors can hinder robust generalization, as their failure in out-of-domain distributions may trigger a cascading failure in subsequent steps; see Sec. 3.2.3. In addition, despite the high generalizability of BUFFER, we observed that during cross-domain testing, users had to manually specify the optimal voxel size for the test data, which hinders fully zero-shot inference.

Under these circumstances, we revisit the generalization problem in point cloud registration and explore how to achieve zero-shot registration while preserving the key benefits of BUFFER's scale normalization strategy. In addition, we remove certain modules that hinder robustness and introduce an adaptive mechanism that determines the voxel size and search radii depending on the given cloud points pair. To the best of our knowledge, this is the first approach to evaluate the zero-shot generalization across diverse scenes covering various environments, geographic regions, scales, sensor types, and acquisition setups.



(a) (b)

Fig. 2: (a) Variation in the number of points after voxelization with different voxel sizes v across datasets. Even in indoor scenes, point counts vary significantly depending on the sensor type (*i.e.,* `TIERS` [59] vs. `3DMatch` [87]). Notably, `TIERS` and `KITTI` [27], both using omnidirectional LiDARs, yield different point densities due to indoor vs. outdoor environments. (b) Empirical distribution of the datasets' maximum range.

3. Preliminaries

3.1. Problem statement

The goal of point cloud registration is to estimate the relative 3D rotation matrix $\boldsymbol{R} \in \mathrm{SO}(3)$ and translation vector $\boldsymbol{t} \in \mathbb{R}^3$ between two unordered 3D point clouds $\mathcal{P}$ and $\mathcal{Q}$. To this end, most approaches [31, 40] follow three steps: a) apply voxel sampling $f_v(\cdot)$ to the point cloud with voxel size v as preprocessing, b) establish associations (or *correspondences*) $\mathcal{A}$, and c) estimate $\boldsymbol{R}$ and $\boldsymbol{t}$.

Formally, by denoting corresponding points for a correspondence (i, j) in $\mathcal{A}$ as $\boldsymbol{p}_i \in f_v(\mathcal{P})$ and $\boldsymbol{q}_j \in f_v(\mathcal{Q})$, respectively, the objective function used for pose estimation can be defined as:

$$\hat{\boldsymbol{R}}, \hat{\boldsymbol{t}} = \underset{\boldsymbol{R}\in\mathrm{SO}(3), \boldsymbol{t}\in\mathbb{R}^3}{\arg\min} \sum_{(i,j)\in\mathcal{A}} \rho\Big(\|\boldsymbol{q}_j - \boldsymbol{R}\boldsymbol{p}_i - \boldsymbol{t}\|_2\Big), \quad (1)$$

where $\rho(\cdot)$ represents a nonlinear kernel function that mitigates the effect of spurious correspondences in $\mathcal{A}$.

3.2. Key observations

If $\mathcal{A}$ in (1) is accurate, solving (1) is easy. However, we have observed there exist three factors that cause learning-based registration to struggle in estimating $\mathcal{A}$ when given out-of-domain data.

3.2.1 Voxel size and search radius

First, dependencies of optimal search radius r for local descriptors and voxel size v for each dataset are problematic. The optimal parameters vary significantly across datasets due to differences in scale and point density (*e.g.,* small indoor scenes vs. large outdoor spaces [5]) Consequently, improper r or v can severely degrade registration performance by failing to account for specific scale and density characteristics of a given environment or sensor; see Sec. 5.1. For instance, in Fig. 2(a), as v controls the maximum number of points that can be fed into the network, a too-small v can trigger out-of-memory errors when outdoor data processed with parameters optimized for indoor environments are taken as input to the network.

In particular, most methods heavily depend on manual tuning, which hinders generalization. Therefore, we employ a *geometric bootstrapping* to adaptively determine v and r at test time based on the scale and point density of the given input clouds; see Sec. 4.1.

3.2.2 Input scale normalization

Next, directly feeding raw x, y, and z values into the network leads to strong in-domain dependency [31, 58]. That is, when a model fits to the training distribution, large scale discrepancies between training and unseen data can cause catastrophic failure (see Fig. 2(b) for an example of maximum range discrepancy). For this reason, we conclude that normalizing input points within local neighborhoods (or *patches)* is necessary to achieve generalizability, ensuring that their coordinates lie within a bounded range (*e.g.,* $[-1, 1]$) [5, 6].

Based on these insights, we adopt patch-based descriptor generation as our pipeline for descriptor matching; see Sec. 4.2.

3.2.3 Keypoint detection

Following Sec. 3.2.2, we observed that point-wise feature extractor modules in existing methods [31, 58, 79] are empirically brittle to out-of-domain data. Because failed keypoint detection leads to the selection of unreliable and non-repeatable points as keypoints, it results in low-quality descriptors and ultimately degrades the quality of $\mathcal{A}$ [29].

An interesting observation is that replacing the learning-based detector with the farthest point sampling (FPS) preserves registration performance. For this reason, we adopt FPS over a learning-based module (see Sec. 5.3). Specifically, we apply it separately at local, middle, and global scales to account for multi-scale variations.

4. BUFFER-X

Building upon our key observations in Sec. 3.2, we present our multi-scale zero-shot registration pipeline; see Fig. 3. First, the appropriate voxel size and radii for each cloud pair are predicted by geometric bootstrapping (Sec. 4.1), considering the overall distribution of cloud points and the density of neighboring points, respectively. Then, we extract Mini-SpinNet-based features [5] for the sampled points via FPS at multiple scales (Sec. 4.2). Finally, at the intra- and cross-scale levels, refined correspondences are estimated based on consensus maximization [66, 67, 88] (Sec. 4.3) and serve as input for the final relative pose estimation using a solver.

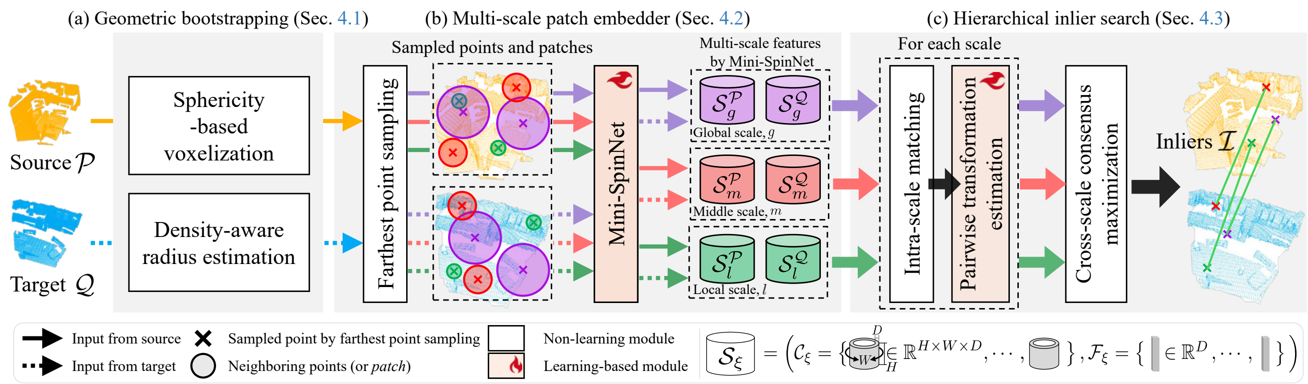


Fig. 3: Overview of our *BUFFER-X*, which mainly consists of three steps. (a) Geometric bootstrapping (Sec. 4.1) to determine the appropriate voxel size and radii for the given source $\mathcal{P}$ and target $\mathcal{Q}$ clouds. (b) Multi-scale patch embedder (Sec. 4.2) to generate patch-wise descriptor $\mathcal{S}_\xi$ for multiple scale $\xi \in \{l, m, g\}$, where l, m, and g represent local, middle, and global scales, respectively. Specifically, Mini-SpinNet [5] outputs cylindrical feature maps $\mathcal{C}_\xi$ and vector feature set $\mathcal{F}_\xi$. (c) Hierarchical inlier search (Sec. 4.3), which first performs nearest neighbor-based intra-scale matching using $\mathcal{F}_\xi^{\mathcal{P}}$ and $\mathcal{F}_\xi^{\mathcal{Q}}$ at each scale, followed by pairwise transformation estimation. Finally, it identifies globally consistent inliers $\mathcal{I}$ across all scales to refine correspondences based on consensus maximization [67, 88].

4.1. Geometric bootstrapping

Sphericity-based voxelization. First, we determine proper voxel size v by leveraging sphericity, quantified using eigenvalues [1, 28], to reflect how the cloud points are dispersed in space. To this end, we apply principal component analysis (PCA) [41] to the covariance of sampled points, which can efficiently capture point dispersion by analyzing eigenvalues while remaining computationally lightweight.

Formally, let $h(\mathcal{P}, \mathcal{Q})$ be a function that selects the larger point cloud based on cardinality, let $g(\mathcal{P}, \delta)$ be a function that samples $\delta\%$ of points from a given point cloud, and let $\boldsymbol{C} \in \mathbb{R}^{3\times 3}$ be the covariance of $g(h(\mathcal{P}, \mathcal{Q}), \delta_v)$, where δ_v is a user-defined sampling ratio. Then, using PCA, three eigenvalues λ_a and their corresponding eigenvectors $\boldsymbol{v}_a$ are calculated as follows:

$$\boldsymbol{C}\boldsymbol{v}_a = \lambda_a \boldsymbol{v}_a, \quad a \in \{1, 2, 3\}, \tag{2}$$

which are assumed to be $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Then, using these properties, we can compute the *sphericity* $\frac{\lambda_3}{\lambda_1}$ [1], which quantifies how evenly a point cloud is distributed in space. Since LiDAR points are primarily distributed along the sensor's horizontal plane (*i.e.,* forming a disc-like shape), $\frac{\lambda_3}{\lambda_1}$ tends to be low compared to RGB-D point clouds.

In addition, as observed in Fig. 2(a), LiDAR point clouds require a larger voxel size; thus, we set v as follows:

$$v = \begin{cases} \kappa_{\text{spheric}}\sqrt{s}, & \text{if } \frac{\lambda_3}{\lambda_1} \geq \tau_v, \\ \kappa_{\text{disc}}\sqrt{s}, & \text{otherwise}, \end{cases} \tag{3}$$

where κ_{spheric} and κ_{disc} are constant user-defined coefficients across all datasets, satisfying $\kappa_{\text{spheric}} < \kappa_{\text{disc}}$, τ_v is a user-defined threshold, and s is the length that represents the spread of points along the eigenvector corresponding to the smallest eigenvalue $\boldsymbol{v}_3$ (*i.e.,* $s = \max(\mathcal{P}_{\text{sampled}} \cdot \boldsymbol{v}_3) - \min(\mathcal{P}_{\text{sampled}} \cdot \boldsymbol{v}_3)$). Consequently, as $\frac{\lambda_3}{\lambda_1}$ and s adapt based on the environment (*i.e.,* indoor or outdoor) and the field of view of the sensor type (*i.e.,* RGB-D or LiDAR point cloud), (3) enables the adaptive setting of v.

Hereafter, for brevity, we denote $f_v(\mathcal{P})$ and $f_v(\mathcal{Q})$ simply as $\mathcal{P}$ and $\mathcal{Q}$, respectively,

Density-aware radius estimation. Next, in contrast to some state-of-the-art approaches [5, 6] that use a single fixed user-defined search radius, we determine r at local, middle, and global scales, respectively, by considering the input point densities. Let neighboring search function within the radius r around a query point $\boldsymbol{p}_q$ be:

$$\mathcal{N}(\boldsymbol{p}_q, \mathcal{P}, r) = \{\boldsymbol{p} \in \mathcal{P} \mid \|\boldsymbol{p} - \boldsymbol{p}_q\|_2 \leq r\}. \tag{4}$$

Then, as presented in Fig. 4(a), the radius for patch-wise descriptor generation for each scale r_ξ is defined as follows:

$$r_\xi = \underset{r}{\arg\min} \left| \frac{1}{N} \sum_{\boldsymbol{p}_q \in \mathcal{P}_r} \text{card}\Big(\mathcal{N}(\boldsymbol{p}_q, \mathcal{P}_r, r)\Big) - \tau_\xi \right|, \tag{5}$$

where $\xi \in \{l, m, g\}$ denotes the scale level (*i.e.,* local, middle, and global scale, respectively), τ_ξ denotes the user-defined threshold, which represents the desired neighborhood density (*i.e.,* average proportion of neighboring points relative to the total number of points), satisfying $\tau_l \leq \tau_m \leq \tau_g$ (accordingly, $r_l \leq r_m \leq r_g$ as presented in Fig. 4(a)), and $\mathcal{P}_r$ is a set of N_r points sampled from $h(\mathcal{P}, \mathcal{Q})$, where N_r is a user-defined parameter for radius estimation. To account for cases where the points are too sparse, we set the maximum truncation radius $r_{\max}$ as $r_\xi \leftarrow \max(r_\xi, r_{\max})$.

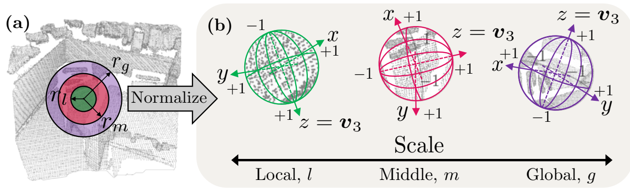


Fig. 4: (a) Visual description of local (r_l), middle (r_m), and global (r_g) radii for the same point to illustrate scale differences and (b) normalized patches ranging from $[-1, 1]$. Note that their reference frames follow the eigenvectors obtained from principal component analysis (PCA) [1, 41]. The z-axis is assigned to the eigenvector $\boldsymbol{v}_3$, which corresponds to the smallest eigenvalue.

4.2. Multi-scale patch embedder

Next, with the voxelized point clouds $\mathcal{P}$ and $\mathcal{Q}$ and radii estimated by (5), patch-wise descriptors are generated at each scale.

Farthest point sampling. As discussed in Sec. 3.2.3, we sample $\mathcal{P}_\xi$ from $\mathcal{P}$ at each scale using FPS to be free from a learning-based keypoint detector (resp. $\mathcal{Q}_\xi$ from $\mathcal{Q}$). Note that instead of extracting local, middle, and global-scale descriptors for the same sampled point [84], we independently sample separate points for each scale, as illustrated in Fig. 3(b). This is because we empirically found that different regions may require distinct scales for optimal feature extraction; see Sec. 5.3.

Mini-SpinNet-based descriptor generation. Using multiple radii r_ξ, we sample patches at three distinct scales, providing a more comprehensive multi-scale representation. Then, we use Mini-SpinNet [5] for descriptor generation, which is a lightweight version of SpinNet [6].

In particular, building on the insights from Sec. 3.2.2, we ensure the scale of points in each patch is normalized to a bounded range of $[-1, 1]$ by dividing by r_ξ; see Fig. 4(b). By doing so, we can resolve the dependency on the in-domain scale. To maintain consistency across patches at all scales, we fix the patch size to N_{patch} and randomly sample when a patch exceeds this size, ensuring a consistent number of points regardless of scale variations.

Finally, taking these normalized patches as inputs, Mini-SpinNet outputs a superset $\mathcal{S}_\xi^{\mathcal{P}}$ consisting of D-dimensional feature vectors $\mathcal{F}_\xi^{\mathcal{P}}$ and cylindrical feature maps $\mathcal{C}_\xi^{\mathcal{P}}$, where corresponds to $\mathcal{P}_\xi$ (resp. $\mathcal{S}_\xi^{\mathcal{Q}}$ consisting of $\mathcal{F}_\xi^{\mathcal{Q}}$ and $\mathcal{C}_\xi^{\mathcal{Q}}$ from $\mathcal{Q}_\xi$), as described in Fig. 3. Note that while BUFFER [5] utilizes learned reference axes to extract cylindrical coordinates, our approach defines the reference axes for each patch by applying PCA to the covariance of points within the patch, setting the z-direction as $\boldsymbol{v}_3$ (as in (2) and illustrated by $z = \boldsymbol{v}_3$ in Fig. 4(b)), to eliminate potential dataset-specific inductive biases.

4.3. Hierarchical inlier search

Here, we first perform inter-scale matching to get initial correspondences $\mathcal{A}_\xi$ at each scale and then establish cross-scale consistent correspondences in a consensus maximization manner.

Intra-scale matching. First, we perform nearest neighbor-based mutual matching [46] between $\mathcal{F}_\xi^{\mathcal{P}}$ and $\mathcal{F}_\xi^{\mathcal{Q}}$, yielding matched correspondences $\mathcal{A}_\xi$ at each scale. Using $\mathcal{A}_\xi$, we extract the corresponding elements from $\mathcal{C}_\xi^{\mathcal{P}}$ and $\mathcal{C}_\xi^{\mathcal{Q}}$, denoted as $\widehat{\mathcal{C}}_\xi^{\mathcal{P}}$ and $\widehat{\mathcal{C}}_\xi^{\mathcal{Q}}$, and the sampled keypoints from $\mathcal{P}_\xi$ and $\mathcal{Q}_\xi$ as $\widehat{\mathcal{P}}_\xi$ and $\widehat{\mathcal{Q}}_\xi$, respectively (*i.e.*, $|\mathcal{A}_\xi| = |\widehat{\mathcal{C}}_\xi^{\mathcal{P}}| = |\widehat{\mathcal{C}}_\xi^{\mathcal{Q}}| = |\widehat{\mathcal{P}}_\xi| = |\widehat{\mathcal{Q}}_\xi|$).

Pairwise transformation estimation. Next, using each cylindrical feature pair $\boldsymbol{c}^{\boldsymbol{p}} \in \widehat{\mathcal{C}}_\xi^{\mathcal{P}}$ and $\boldsymbol{c}^{\boldsymbol{q}} \in \widehat{\mathcal{C}}_\xi^{\mathcal{Q}}$ at each scale, each of whose size is $\mathbb{R}^{H \times W \times D}$, we calculate pairwise 3D relative transformation between two patches. Here, H, W, and D denote the height, sector size for the yaw direction along the z-axis of the reference axes, and feature dimensionality of a cylindrical feature, respectively.

As mentioned earlier, since the cylindrical feature is aligned with the local reference axes via PCA, the relative 3D rotation between $\boldsymbol{v}_3^{\boldsymbol{p}}$ (resp. $\boldsymbol{v}_3^{\boldsymbol{q}}$) and the unit z-axis, $\boldsymbol{z} = [0\ 0\ 1]^\intercal$, can be calculated using Rodrigues' rotation formula [47] as follows:

$$\boldsymbol{R}^{\boldsymbol{p}} = \boldsymbol{I} + \sin(\theta^{\boldsymbol{p}})[\boldsymbol{n}^{\boldsymbol{p}}]_\times + \big(1 - \cos(\theta^{\boldsymbol{p}})\big)[\boldsymbol{n}^{\boldsymbol{p}}]_\times^2, \quad (6)$$

where $\boldsymbol{n}^{\boldsymbol{p}} = \boldsymbol{v}_3^{\boldsymbol{p}} \times \boldsymbol{z}$, $\theta^{\boldsymbol{p}} = \cos^{-1}(\boldsymbol{v}_3^{\boldsymbol{p}} \cdot \boldsymbol{z})$, and $[\cdot]_\times$ denotes the skew operator (resp. $\boldsymbol{R}^{\boldsymbol{q}}$). Thus, once the yaw rotation between the two patches $\boldsymbol{R}_{\text{yaw}}$ is determined, the full 3D rotation can be obtained as $\boldsymbol{R} = (\boldsymbol{R}^{\boldsymbol{q}})^\intercal \boldsymbol{R}_{\text{yaw}} \boldsymbol{R}^{\boldsymbol{p}}$.

As explained by Ao *et al.* [5], $\boldsymbol{c}^{\boldsymbol{p}}$ and $\boldsymbol{c}^{\boldsymbol{q}}$ follow discretized SO(2)-equivariant representation; thus, by finding the yaw rotation that maximizes circular cross-correlation between $\boldsymbol{c}^{\boldsymbol{p}}$ and $\boldsymbol{c}^{\boldsymbol{q}}$, we can estimate the relative SO(2) rotation $\boldsymbol{R}_{\text{yaw}}$. To this end, a 4D matching cost volume $\boldsymbol{V} \in \mathbb{R}^{H \times W \times W \times D}$ is constructed to represent the sector-wise differences between $\boldsymbol{c}^{\boldsymbol{p}}$ and $\boldsymbol{c}^{\boldsymbol{q}}$. Then, $\boldsymbol{V}$ is processed by a 3D cylindrical convolutional network (3DCCN) [6], mapping $\boldsymbol{V}$ to a score vector $\boldsymbol{\beta}$ of size W.

By applying the softmax operation $\sigma(\cdot)$ to $\boldsymbol{\beta}$, we obtain $\sigma(\boldsymbol{\beta})$, where the w-th element $\sigma_w(\boldsymbol{\beta}) \in [0, 1]$ represents the probability mass assigned to the discrete yaw rotation index w. Using this distribution, the discrete rotation offset d is computed as follows:

$$d = \sum_{w=1}^{W} \sigma_w(\boldsymbol{\beta}) \times w. \quad (7)$$

Finally, $\boldsymbol{R}_{\text{yaw}}$ is calculated as follows:

$$\boldsymbol{R}_{\text{yaw}} = \begin{bmatrix} \cos\left(\frac{2\pi d}{W}\right) & -\sin\left(\frac{2\pi d}{W}\right) & 0 \\ \sin\left(\frac{2\pi d}{W}\right) & \cos\left(\frac{2\pi d}{W}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Subsequently, the translation vector is given by $\boldsymbol{t} = \boldsymbol{q} - \boldsymbol{R}\boldsymbol{p}$, where $\boldsymbol{p} \in \widehat{\mathcal{P}}_\xi$ and $\boldsymbol{q} \in \widehat{\mathcal{Q}}_\xi$ are a matched point pair.

Cross-scale consensus maximization. Then, using per-pair $(\boldsymbol{R}, \boldsymbol{t})$ estimates from all scales, the 3D point pairs with the largest cardinality across scales should be selected as the final inlier correspondences $\mathcal{I}$, ensuring cross-scale consistency. To achieve this, we formulate the cross-scale inlier selection as *consensus maximization* problem [67, 88].

Formally, by denoting $N = \sum_\xi |\mathcal{A}_\xi|$, let $(\boldsymbol{R}, \boldsymbol{t}) \in \mathcal{T}$ be a candidate transformation set of size N, and let $(\boldsymbol{p}_n, \boldsymbol{q}_n) \in \mathcal{D}$ be the set of matched point pairs, where $n \in \{1, \ldots, N\}$, $\boldsymbol{p}_n \in \bigcup_\xi \widehat{\mathcal{P}}_\xi$ and $\boldsymbol{q}_n \in \bigcup_\xi \widehat{\mathcal{Q}}_\xi$. Then, $\mathcal{I}$ is estimated as follows:

$$\max_{(\boldsymbol{R},\boldsymbol{t}) \in \mathcal{T},\, \mathcal{I}} |\mathcal{I}| \tag{9}$$
$$\text{s.t. } \|\boldsymbol{R}\boldsymbol{p}_n + \boldsymbol{t} - \boldsymbol{q}_n\|_2 < \epsilon, \;\; \forall (\boldsymbol{p}_n, \boldsymbol{q}_n) \in \mathcal{I} \subseteq \mathcal{D},$$

where ϵ is an inlier threshold.

Finally, $\mathcal{I}$ is given as input to a solver, such as RANSAC [26] or TEASER++ [76] to estimate $\hat{\boldsymbol{R}}$ and $\hat{\boldsymbol{t}}$. For a fair comparison with existing approaches, we use RANSAC.

4.4. Loss function and training

Loss functions. Unlike BUFFER, which was trained in four stages, our network follows a relatively simpler two-stage training process thanks to its detector-free nature. First, we train the feature discriminability of Mini-SpinNet descriptors using contrastive learning [81], followed by training d in (7) to improve transformation estimation accuracy.

In particular, we employ the Huber loss [91] $\rho_{\text{Huber}}(\cdot)$ for training d to remain robust to outliers [9], while balancing sensitivity to small errors, which is formulated as follows:

$$\rho_{\text{Huber}}(r) = \begin{cases} \frac{1}{2}r^2, & \text{if } |r| \leq \delta \\ \delta(|r| - \frac{1}{2}\delta), & \text{otherwise,} \end{cases} \tag{10}$$

where r denotes the residual and δ denotes the user-defined truncation threshold. Then, denoting the total number of data pairs by N_d, the γ-th predicted offset by d_γ, and the corresponding ground-truth offset by d^*_γ, the loss function $\mathcal{L}_d$ is defined as follows:

$$\mathcal{L}_d = \frac{1}{N_d} \sum_{\gamma=1}^{N_d} \rho_{\text{Huber}}(d_\gamma - d^*_\gamma). \tag{11}$$

Patch distribution augmentation. Furthermore, we propose an inter-patch point distribution augmentation to allow Mini-SpinNet to experience a wider variety of patch distribution patterns. Specifically, we empirically sample the radius within $[\frac{2}{3}r, \frac{4}{3}r]$ based on a uniform probability. As mentioned in Sec. 4.2, since N_{patch} points within the radius are randomly selected as an input, a diverse set of patterns can be provided as r varies.

Notably, training is conducted using only a single scale. This leverages the scale normalization characteristic of BUFFER-X, making it unnecessary to train with multi-scale separately.

5. Experiments

Datasets. As presented in Table A2, we designed our generalizability benchmark using eleven different datasets [27, 31, 33, 55, 59, 60, 68, 70, 80, 87] to ensure balanced consideration of the following aspects: a) variation in environmental scales (*i.e.,* indoor and outdoor environments), b) different scanning patterns with different sensor types, c) acquisition setups, and d) diversity of geographic and cultural environments as the data was collected across Europe, Asia, and the USA (*i.e.,* `Oxford`, `KAIST`, and `MIT` campuses, respectively). More details can be found in Appendices B and C.

Training settings. Then, we only train the network on a single dataset, such as `3DMatch` [87] or `KITTI` [27]. Using the same hyperparameters of BUFFER [5], we conducted a two-stage optimization (*i.e.,* Mini-SpinNet is first trained, followed by training the 3DCCN) and we used Adam optimizer [34] with a learning rate of 0.001, a weight decay of 1e-6, and a learning rate decay of 0.5. We used NVIDIA GeForce RTX 3090 with AMD EPYC 7763 64-Core.

Testing settings. In the case of the existing datasets [27, 31, 55, 68, 87], we follow the conventional given pairs. The description of the newly employed datasets for evaluation can be found in Appendix B.

Evaluation Metrics. As a key metric, we use the success rate, which directly assesses the robustness of global registration [39]. Specifically, a registration is deemed successful if the translation and rotation errors are within τ_{trans} and τ_{rot}, respectively [81]. For successful cases, we evaluated the performance using relative translation error (RTE) and relative rotation error (RRE), which are defined as follows:

- RTE $= \sum_{n=1}^{N_{\text{success}}} (\boldsymbol{t}_{n,\text{GT}} - \hat{\boldsymbol{t}}_n)^2 / N_{\text{success}}$,
- RRE $= \frac{180}{\pi} \sum_{n=1}^{N_{\text{success}}} |\cos^{-1}(\frac{\text{Tr}(\hat{\boldsymbol{R}}_n^\intercal \boldsymbol{R}_{n,\text{GT}})-1}{2})| / N_{\text{success}}$

where $\boldsymbol{t}_{n,\text{GT}}$ and $\boldsymbol{R}_{n,\text{GT}}$ denote the n-th ground truth translation and rotation, respectively; N_{success} represents the number of successful registration. The more detailed criteria for determining a successful registration are provided in table A2.

Env.		Indoor					Outdoor						Average rank
	Dataset	3DMatch	3DLoMatch	ScanNet++i	ScanNet++F	TIERS	KITTI	WOD	KAIST	MIT	ETH	Oxford	
Conven-tional	FPFH [62] + FGR [93] + 👁	62.53	15.42	77.68	92.31	80.60	98.74	**100.00**	89.80	74.78	91.87	99.00	9.55
	FPFH [62] + Quatro [42] + 👁	8.22	1.74	9.88	97.27	86.57	99.10	**100.00**	91.46	79.57	51.05	91.03	10.73
	FPFH [62] + TEASER++ [76] + 👁	52.00	13.25	66.15	97.22	73.13	98.92	**100.00**	89.20	71.30	93.69	99.34	10.00
Deep learning-based	FCGF [21]	88.18	40.09	72.90	88.69	55.96	0.00	0.00	0.00	0.00	54.98	0.00	15.00
	+ 👁	88.18	40.09	85.87	88.69	78.62	90.27	97.69	92.91	92.61	54.98	93.68	10.18
	+ 👁 + ⤢	88.18	40.09	85.87	88.69	80.11	94.41	97.69	93.55	93.04	55.53	95.68	9.55
	Predator [31]	90.60	62.40	75.94	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	15.73
	+ 👁	90.60	62.40	75.94	29.81	56.44	0.00	0.00	0.95	0.00	0.14	0.33	14.55
	+ 👁 + ⤢	90.60	62.40	75.94	86.01	75.74	77.29	86.92	87.09	79.56	54.42	93.68	11.82
	GeoTransformer [58]	92.00	75.00	91.18	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	14.00
	+ 👁	92.00	75.00	91.18	7.54	5.06	0.36	0.77	0.25	0.87	0.00	0.33	13.09
	+ 👁 + ⤢	92.00	75.00	92.72	97.02	92.99	92.43	89.23	91.86	95.65	71.53	97.01	6.27
	BUFFER [5]	92.90	71.80	92.72	93.75	62.30	0.00	1.54	0.50	6.96	97.62	0.66	10.45
	+ 👁	92.90	71.80	93.01	94.69	88.96	99.46	**100.00**	97.24	95.65	99.30	99.00	3.82
	+ 👁 + ⤢	92.90	71.80	93.01	94.69	88.96	99.46	**100.00**	97.24	95.65	99.30	99.00	3.82
	PARENet [79]	95.00	**80.50**	90.84	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	13.27
	+ 👁	95.00	**80.50**	90.84	43.75	6.21	0.18	0.77	0.75	1.30	1.40	1.66	11.55
	+ 👁 + ⤢	95.00	**80.50**	90.84	87.95	75.06	84.86	92.31	86.44	84.78	69.42	93.36	8.82
	Ours with only r_m	93.38	71.69	93.10	99.60	90.80	**99.82**	**100.00**	99.05	95.65	99.30	99.34	3.00
	Ours	**95.58**	74.18	**94.99**	**99.90**	**93.45**	**99.82**	**100.00**	**99.15**	**97.39**	**99.72**	**99.67**	**1.55**

Table 1: Quantitative comparison of generalization performance in terms of success rate (%). Deep learning-based models were trained only on 3DMatch [87] and RANSAC was used with a maximum iteration of 50K. The icons represent oracle tuning (👁) for voxel size and radius, and scale alignment (⤢) to normalize dataset scales to be similar to that of 3DMatch data (*e.g.,* the scale of KITTI, which typically uses a voxel size of 0.3 m, is adjusted to match the scale of 3DMatch, where 0.025 m is commonly used, by dividing by $\frac{0.3}{0.025}$).

5.1. Analyses on the generalization

First, we demonstrate that existing methods struggle in achieving out-of-the-box generalization, leading to performance degradation due to the issues explained in Sec. 3. In this experiment, we mainly used renowned learning-based approaches: FCGF [21], Predator [31], GeoTransformer (*GeoT* for brevity) [58], BUFFER [5], and PARENet [79].

As shown in Table 1, models trained with small voxel sizes and search radii for indoor datasets exhibited substantial performance degradation in outdoor scenarios. In particular, as explained in Sec. 3.2.1, we observed that some approaches did not even work due to out-of-memory issues (*i.e.,* N/A in Table 1) caused by an excessively large number of input points.

Once a properly user-tuned voxel size and search radius were provided (referred to as *oracle tuning*, 👁), BUFFER showed a remarkable performance increase owing to its patch-wise input scale normalization characteristics. In contrast, other approaches still showed relatively lower success rates because the networks received point clouds with magnitudes not encountered during training. This potential limitation is further evidenced by the performance improvement of Predator after scale alignment, supporting our claim that scale normalization is a key factor in achieving generalizability.

5.2. Performance comparison

Second, we demonstrate that our proposed algorithm, inspired by these key observations, achieves substantial out-of-the-box generalization; see Table 1. In particular, our approach achieved lower RTE and RRE than state-of-the-art approaches even with oracle tuning and scale alignment (Fig. 5), while maintaining competitive in-domain performance (Table 2). Therefore, this experimental evidence supports our claim that our algorithm achieves a high generalization capability and successfully performs in-domain scenarios.

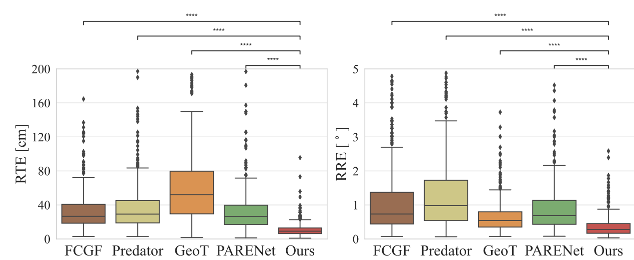


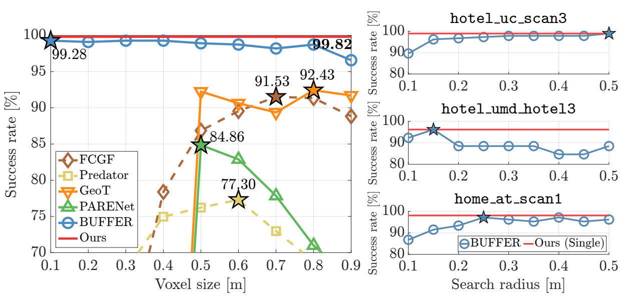
Fig. 5: Relative translation error (RTE) and relative rotation error (RRE) of our approach to state-of-the-art methods, all trained on 3DMatch and tested on KITTI, with oracle tuning and scale alignment, corresponding to those in Table 1 under the + 👁 + ⤢ setting. The **** annotations indicate measurements with a p-value $< 10^{-4}$ after a paired t-test.

5.3. Ablation study

Impact of geometric bootstrapping. Fig. 6 supports our claim that voxel size and radius have a more direct impact on performance than expected. Unlike BUFFER, which relies on manual tuning, our method automatically estimates optimal v and r, adapting to different scenes and maintaining consistency across varying dataset densities (Fig. 6(b)).

	Method	RTE [cm] ↓	RRE [°] ↓	Succ. rate [%] ↑
Conventional	G-ICP [65]	8.56	**0.22**	37.95
	FPFH [62] + FGR [93]	18.75	0.38	98.74
	FPFH [62] + Quatro [42]	18.56	0.93	99.10
	FPFH [62] + TEASER++ [76]	15.35	0.68	98.92
	KISS-Matcher [40]	21.33	0.96	99.46
Learning-based	3DFeat-Net [81]	25.90	0.57	95.97
	FCGF [21]	6.47	0.23	98.92
	DIP [52]	8.69	0.44	97.30
	Predator [31]	5.60	0.24	**99.82**
	SpinNet [6]	9.88	0.47	99.10
	CoFiNet [84]	8.20	0.41	**99.82**
	D3Feat [8]	11.00	0.24	**99.82**
	GeDi [53]	7.55	0.33	**99.82**
	GeoTransformer [58]	7.40	0.27	**99.82**
	BUFFER [5]	7.46	0.26	99.64
	PARE-Net [79]	**4.90**	0.23	**99.82**
	Ours	7.74	0.27	**99.82**

Table 2: In-domain quantitative results in terms of relative translation error (RTE), relative rotation error (RRE), and success rate on `KITTI` [81].



(a) (b)

Fig. 6: Effect of voxel size and search radius on success rates, where ★ indicates the best performance after tuning. (a) Impact of voxel size (with ⌕) on models trained on `3DMatch`, evaluated on `KITTI`. (b) BUFFER-X vs. BUFFER across different radii, showing that optimal radius may vary within the same dataset.

Learning-based detector vs. Farthest point sampling. Interestingly, FPS rather showed better performance than the learning-based keypoint detector in BUFFER [5], even in its training domain (*i.e.,* in `3DMatch` and `3DLoMatch`). In particular, the performance gap sometimes becomes more pronounced in the out-of-domain scenes, demonstrating that robust cross-domain generalization can be achieved without the need for additional learning-based keypoint selection.

Impact of multi-scale. While we demonstrated that using only the middle scale in a single-scale setting is comparable to the multi-scale approach (see Table 1), Table 3 shows that incorporating multiple scales further increases the success rate. This implies that correspondences across scales complement each other, leading to higher success rates. However, increasing the number of scales introduces a trade-off between accuracy and computational cost (Fig. 8), allowing users to balance efficiency and performance based on their specific needs.

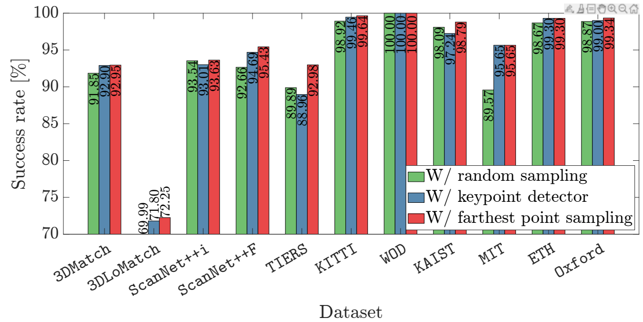


Fig. 7: Comparison of success rates between random sampling, learning-based keypoint detection in BUFFER [5], and our farthest point sampling (FPS) strategy, showing that FPS performs comparably or even better across various datasets.

Local	Middle	Global	RTE [cm] ↓	RRE [°] ↓	Succ. rate [%] ↑	Hz ↑
✓			6.57	2.15	84.06	**5.61**
	✓		5.87	1.85	93.38	5.47
		✓	6.06	1.91	93.57	5.49
✓	✓		**5.73**	1.81	94.31	2.35
✓		✓	5.77	1.81	94.02	2.36
	✓	✓	5.78	1.81	94.62	2.33
✓	✓	✓	5.78	**1.79**	**95.58**	1.81

Table 3: Ablation study: the impact of different scale combinations on registration performance in the `3DMatch` [87].

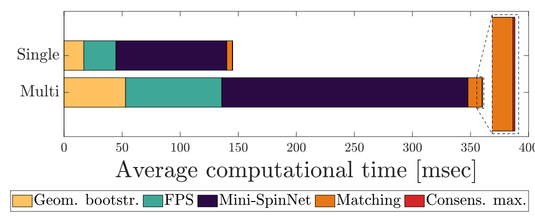


Fig. 8: Area plot of computation time per scale for each module on an NVIDIA GeForce RTX 3090 using `3DMatch` [87].

5.4. Limitations

As seen in Table 1, our approach showed lower success rate in `3DLoMatch`, which only have 10-30% overlaps. This is because Eq. (9) selects correspondences with the largest cardinality as inliers. However, in partial overlap scenarios, maximizing the number of correspondences might not yield the actual global optimum (*i.e.,* there might exist $\mathcal{I}^*$ that satisfies $|\mathcal{I}^*| \leq |\mathcal{I}|$ but leads to a better relative pose estimate). This highlights a trade-off between generalization and robustness to partial overlaps (see Appendix D for further analyses).

6. Conclusion

In this study, we addressed the generalization limitations of deep learning-based registration and analyzed key factors hindering it. Based on these insights, we proposed a fully zero-shot pipeline, *BUFFER-X*, and introduced a comprehensive benchmark for evaluating generalization on real-world point cloud data. In future works, we plan to study how to boost the inference speed for better usability.

Acknowledgments

This work was supported by IITP grant (RS-2021-II211343: AI Graduate School Program at Seoul National University) (5%), and by NRF grant funded by the Korea government (MSIT) (No. 2023R1A1C200781211 (65%) and No. RS-2024-00461409 (30%), respectively).

References

[1] Evangelos Alexiou, Xuemei Zhou, Irene Viola, and Pablo Cesar. PointPCA: Point cloud objective quality assessment using PCA-based descriptors. *EURASIP Journal on Image and Video Processing*, 2024(1):20, 2024. 4, 5

[2] Sheng Ao, Yulan Guo, Shangtai Gu, Jindong Tian, and Dong Li. SGHs for 3D local surface description. *IET Computer Vision*, 14(4):154–161, 2020. 1

[3] Sheng Ao, Yulan Guo, Qingyong Hu, Bo Yang, Andrew Markham, and Zengping Chen. You only train once: Learning general and distinctive 3d local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3949–3967, 2022. 2

[4] Sheng Ao, Yulan Guo, Jindong Tian, Yong Tian, and Dong Li. A repeatable and robust local reference frame for 3D surface matching. *Pattern Recognition*, 100:107186, 2020. 1

[5] Sheng Ao, Qingyong Hu, Hanyun Wang, Kai Xu, and Yulan Guo. BUFFER: Balancing accuracy, efficiency, and generalizability in point cloud registration. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1255–1264, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[6] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. SpinNet: Learning a general surface descriptor for 3D point cloud registration. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11753–11762, 2021. 1, 2, 3, 4, 5, 8

[7] Koki Aoki, Kenji Koide, Shuji Oishi, Masashi Yokozuka, Atsuhiko Banno, and Junichi Meguro. 3D-BBS: Global localization for 3D point cloud scan matching using branch-and-bound algorithm. *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1796–1802, 2024. 2

[8] 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 *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 8

[9] Jonathan T. Barron. A general and adaptive robust loss function, 2019. 6

[10] Lukas Bernreiter, Lionel Ott, Juan Nieto, Roland Siegwart, and Cesar Cadena. PHASER: A robust and correspondence-free global point cloud registration. *IEEE Robotics and Automation Letters*, 6(2):855–862, 2021. 1, 2

[11] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2), 1992. 2

[12] Mark Brown, David Windridge, and Jean-Yves Guillemaut. A family of globally optimal branch-and-bound algorithms for 2D-3D correspondence-free registration. *Pattern Recognition*, 93:36–54, 2019. 2

[13] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. LCDNet: Deep loop closure detection and point cloud registration for LiDAR SLAM. *IEEE Trans. Robotics*, 38(4):2074–2093, 2022. 2, 4

[14] Guangyan Chen, Meiling Wang, Yi Yang, Li Yuan, and Yufeng Yue. Fast and robust point cloud registration with tree-based transformer. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 773–780, 2024. 1

[15] Hong Chen, Pei Yan, Sihe Xiang, and Yihua Tan. Dynamic cues-assisted transformer for robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21698–21707, 2024. 1, 2

[16] Tianyi Chen, Bo Ji, Tianyu Ding, Biyi Fang, Guanyi Wang, Zhihui Zhu, Luming Liang, Yixin Shi, Sheng Yi, and Xiao Tu. Only train once: A one-shot neural network training and pruning framework. *Advances in Neural Information Processing Systems*, 34:19637–19651, 2021. 1

[17] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Jens Behley, and Cyrill Stachniss. OverlapNet: A siamese network for computing LiDAR scan similarity with applications to loop closing and localization. *Autonomous Robots*, 46(1):61–81, 2022. 2

[18] Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance evaluation of RANSAC family. *J. of Computer Vision*, 24(3):271–300, 1997. 2

[19] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[20] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal Convnets: Minkowski convolutional neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019. 2

[21] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Intl. Conf. on Computer Vision (ICCV)*, pages 8958–8966, 2019. 2, 7, 8, 6

[22] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*, pages 236–243, 2003. 2

[23] Zhen Dong, Bisheng Yang, Yuan Liu, Fuxun Liang, Bijun Li, and Yufu Zang. A novel binary shape context for 3D local surface description. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:431–452, 2017. 2

[24] Alexey Dosovitskiy and Josip Djolonga. You only train once: Loss-conditional training of deep networks. In *International conference on learning representations*, 2019. 1

[25] Kaveh Fathian and Tyler Summers. CLIPPER+: A Fast Maximal Clique Algorithm for Robust Global Registration. *IEEE Robotics and Automation Letters*, 2024. 1

[26] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981. 2, 6

[27] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *Intl. J. of Robotics Research*, 32(11):1231–1237, 2013. 1, 2, 3, 6

[28] Signe Schilling Hansen, Verner Brandbyge Ernstsen, Mikkel Skovgaard Andersen, Zyad Al-Hamdani, Ramona Baran, Manfred Niederwieser, Frank Steinbacher, and Aart Kroon. Classification of boulders in coastal environments using random forest machine learning on topo-bathymetric LiDAR data. *Remote Sensing*, 13(20):4101, 2021. 4

[29] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, pages 10–5244, 1988. 3

[30] R.I. Hartley and F. Kahl. Global optimization through rotation space search. *Intl. J. of Computer Vision*, 82(1):64–79, 2009. 2

[31] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4265–4274, Jun. 2021. 1, 2, 3, 6, 7, 8, 5

[32] Tianyu Huang, Liangzu Peng, Rene Vidal, and Yun-Hui Liu. Scalable 3D Registration via Truncated Entry-wise Absolute Residuals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27477–27487, June 2024. 1

[33] Minwoo Jung, Wooseong Yang, Dongjae Lee, Hyeonjae Gil, Giseop Kim, and Ayoung Kim. HeLiPR: Heterogeneous LiDAR dataset for inter-LiDAR place recognition under spatial and temporal variations. *The International Journal of Robotics Research*, 2023. 1, 2, 6, 4

[34] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[35] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Voxelized GICP for fast and accurate 3D point cloud registration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 11054–11059, 2021. 2

[36] Kanghee Lee, Junha Lee, and Jaesik Park. Learning to register unbalanced point pairs. *arXiv preprint arXiv:2207.04221*, 2022. 2

[37] Huan Lei, Guang Jiang, and Long Quan. Fast descriptors and correspondence propagation for robust global point cloud registration. *IEEE Trans. Image Processing*, 26(8):3614–3623, 2017. 2

[38] Hyungtae Lim, Seoyeon Jang, Benedikt Mersch, Jens Behley, Hyun Myung, and Cyrill Stachniss. Helimos: A dataset for moving object segmentation in 3d point clouds from heterogeneous lidar sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 14087–14094, 2024. 2

[39] Hyungtae Lim, Beomsoo Kim, Daebeom Kim, Eungchang Mason Lee, and Hyun Myung. Quatro++: Robust global registration exploiting ground segmentation for loop closing in LiDAR SLAM. *Intl. J. of Robotics Research*, pages 685–715, 2024. 1, 2, 6

[40] Hyungtae Lim, Daebeom Kim, Gunhee Shin, Jingnan Shi, Ignacio Vizzo, Hyun Myung, Jaesik Park, and Luca Carlone. KISS-Matcher: Fast and Robust Point Cloud Registration Revisited. *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2025. Accepted. To appear. 3, 8

[41] Hyungtae Lim, Minho Oh, and Hyun Myung. Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3D LiDAR sensor. *IEEE Robotics and Automation Letters*, 6(4):6458–6465, 2021. 4, 5

[42] Hyungtae Lim, Suyong Yeon, Soohyun Ryu, Yonghan Lee, Youngji Kim, Jaeseong Yun, Euigon Jung, Donghwan Lee, and Hyun Myung. A single correspondence is enough: Robust global registration to avoid degeneracy in urban environments. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 8010–8017, 2022. 1, 2, 7, 8, 6

[43] Quan Liu, Hongzi Zhu, Zhenxi Wang, Yunsong Zhou, Shan Chang, and Minyi Guo. Extend your own correspondences: Unsupervised distant point cloud registration by progressive distance extension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20816–20826, 2024. 1, 2

[44] Shaocong Liu, Tao Wang, Yan Zhang, Ruqin Zhou, Li Li, Chenguang Dai, Yongsheng Zhang, Longguang Wang, and Hanyun Wang. Deep semantic graph matching for large-scale outdoor point cloud registration. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 1

[45] Xin Liu, Rong Qin, Junchi Yan, and Jufeng Yang. Ncmnet: Neighbor consistency mining network for two-view correspondence pruning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[46] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. of Computer Vision*, 60(2):91–110, 2004. 5

[47] Johan Ernest Mebius. Derivation of the Euler-Rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations. *arXiv preprint math/0701759*, 2007. 5

[48] Juncheng Mu, Lin Bie, Shaoyi Du, and Yue Gao. ColorPCR: Color point cloud registration with multi-stage geometric-color fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21061–21070, 2024. 1

[49] Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(5):783–794, 2009. 2

[50] Jin Pan, Zhe Min, Ang Zhang, Han Ma, and Max Q-H Meng. Multi-view global 2D-3D registration based on branch and bound algorithm. In *Proc. IEEE Int. Conf. Robot. Biomim.*, pages 3082–3087, 2019. 2

[51] Chavdar Papazov, Sami Haddadin, Sven Parusel, Kai Krieger, and Darius Burschka. Rigid 3D geometry matching for grasping of known objects in cluttered scenes. *Intl. J. of Robotics Research*, 31(4):538–553, 2012. 2

[52] Fabio Poiesi and Davide Boscaini. Distinctive 3D local deep descriptors. In *Intl. Conf. on Pattern Recognition (ICPR)*, pages 5720–5727, 2021. 8

[53] Fabio Poiesi and Davide Boscaini. Learning general and distinctive 3D local deep descriptors for point cloud registration. *IEEE Trans. Pattern Anal. Machine Intell.*, 45(3):3979–3985, 2022. 1, 8

[54] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013. 2

[55] François Pomerleau, Ming Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, 2012. 1, 6, 2

[56] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart. Challenging data sets for point cloud registration algorithms. *Intl. J. of Robotics Research*, 31(14):1705–1711, 2012. 2

[57] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 2

[58] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. GeoTransformer: Fast and robust point cloud registration with geometric transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9806–9821, 2023. 2, 3, 7, 8, 6

[59] Li Qingqing, Yu Xianjia, Jorge Pena Queralta, and Tomi Westerlund. Multi-modal LiDAR dataset for benchmarking general-purpose localization and mapping algorithms. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3837–3844, 2022. 1, 2, 3, 6, 4

[60] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. The newer college dataset: Handheld lidar, inertial and vision with ground truth. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4353–4360. IEEE, 2020. 1, 6, 3, 4

[61] Mohammad Rouhani and Angel D Sappa. Correspondence free registration through a point-to-model distance minimization. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2150–2157, 2011. 2

[62] R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3212–3217. Citeseer, 2009. 7, 8, 6

[63] Kwonyoung Ryu, Soonmin Hwang, and Jaesik Park. Instant domain augmentation for lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9350–9360, 2023. 1

[64] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. In *Computer Graphics Forum*, pages 214–226, 2007. 2

[65] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized ICP. In *Robotics: Science and Systems (RSS)*, Jun. 2009. 2, 8

[66] Pengcheng Shi, Shaocheng Yan, Yilin Xiao, Xinyi Liu, Yongjun Zhang, and Jiayuan Li. RANSAC back to SOTA: A two-stage consensus filtering for real-time 3D registration. *IEEE Robotics and Automation Letters*, 2024. 1, 3

[67] Lei Sun and Lu Deng. TriVoC: Efficient voting-based consensus maximization for robust point cloud registration with extreme outlier ratios. *IEEE Robotics and Automation Letters*, 7(2):4654–4661, 2022. 3, 4, 6

[68] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1, 6, 2

[69] 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. 2

[70] Y. Tian, Y. Chang, L. Quang, A. Schang, C. Nieto-Granda, J.P. How, and L. Carlone. Resilient and distributed multi-robot visual SLAM: Datasets, experiments, and lessons learned. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023. 1, 6, 3, 4

[71] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. KISS-ICP: In defense of point-to-point ICP – Simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters*, pages 1029–1036, 2023. 2

[72] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler Faster Stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 2

[73] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 2

[74] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1127–1134, 2020. 5

[75] H. Yang and L. Carlone. A polynomial-time solution for robust registration with extreme outlier rates. In *Robotics: Science and Systems (RSS)*, 2019. 2

[76] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. Robotics*, 37(2):314–333, 2020. 1, 2, 6, 7, 8

[77] J. Yang, H. Li, D. Campbell, and Y. Jia. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. Pattern Anal. Machine Intell.*, 38(11):2241–2254, Nov. 2016. 1, 2

[78] Jiaqi Yang, Xiyu Zhang, Peng Wang, Yulan Guo, Kun Sun, Qiao Wu, Shikun Zhang, and Yanning Zhang. MAC: Maximal Cliques for 3D Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[79] Runzhao Yao, Shaoyi Du, Wenting Cui, Canhui Tang, and Chengwu Yang. Pare-net: Position-aware rotation-equivariant networks for robust point cloud registration. In *European Conference on Computer Vision*, pages 287–303. Springer, 2024. 3, 7, 8, 6

[80] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 1, 6

[81] Z.J. Yew and G.H. Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *European Conf. on Computer Vision (ECCV)*, 2018. 2, 6, 8

[82] Huan Yin, Xuecheng Xu, Sha Lu, Xieyuanli Chen, Rong Xiong, Shaojie Shen, Cyrill Stachniss, and Yue Wang. A survey on global lidar localization: Challenges, advances and open problems. *Intl. J. of Computer Vision*, pages 1–33, 2024. 2

[83] Pengyu Yin, Shenghai Yuan, Haozhi Cao, Xingyu Ji, Shuyang Zhang, and Lihua Xie. Segregator: Global point cloud registration with semantic and geometric cues. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2848–2854, 2023. 1

[84] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. CofiNet: reliable coarse-to-fine correspondences for robust pointcloud registration. *Advances in Neural Information Processing Systems*, 34:23872–23884, 2021. 5, 8

[85] Zhiyuan Yu, Zheng Qin, Lintao Zheng, and Kai Xu. Learning instance-aware correspondences for robust multi-instance point cloud registration in cluttered scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19605–19614, June 2024. 2

[86] Yongzhe Yuan, Yue Wu, Xiaolong Fan, Maoguo Gong, Qiguang Miao, and Wenping Ma. Inlier confidence calibration for point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5312–5321, 2024. 1, 2

[87] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and T Funkhouser. 3DMatch: Learning the matching of local 3d geometry in range scans. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, page 4, 2017. 1, 2, 3, 6, 7, 8

[88] Xinyue Zhang, Liangzu Peng, Wanting Xu, and Laurent Kneip. Accelerating globally optimal consensus maximization in geometric vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4280–4297, 2024. 3, 4, 6

[89] Xiyu Zhang, Jiaqi Yang, Shikun Zhang, and Yanning Zhang. 3D registration with maximal cliques. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17745–17754, 2023. 1

[90] Yifei Zhang, Hao Zhao, Hongyang Li, and Siheng Chen. FastMAC: Stochastic Spectral Sampling of Correspondence Graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17857–17867, 2024. 1

[91] Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15(1):59–76, 1997. 6

[92] Mingyang Zhao, Jingen Jiang, Lei Ma, Shiqing Xin, Gaofeng Meng, and Dong-Ming Yan. Correspondence-free non-rigid point set registration using unsupervised clustering analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21199–21208, 2024. 2

[93] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 766–782, 2016. 1, 2, 7, 8, 6

[94] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948, 2021. 2