Register Anything Model for Generalizable and Robust Point Cloud Registration

Zheng Qin[1] **Junhua Xi**[2,3] **Yan Huang**[1] **Jiahong Lai**[1,2] **Anwen Huang**[1]
Qiong Li[1*] **Guangda Zhang**[1] **Kai Xu**[4*]

[1]Defense Innovation Institute, Academy of Military Sciences, China
[2]College of Computer Science and Technology, National University of Defense Technology, China
[3]College of Data Intelligence, Information Support Force Engineering University, China
[4]Institute of AI for Industry, Chinese Academy of Sciences, China

https://github.com/qinzheng93/RegisterAnything

Abstract

We present Register Anything Model (RAM), a unified model for generalizable 3D point cloud registration that robustly estimates 6-DoF transformations across diverse domains spanning sensors and environments. Existing registration methods are typically tailored to a single domain, with carefully tuned architectures and hyperparameters, and thus degrade markedly when applied to novel domains. We bridge this gap by training a single model on diverse data from indoor, outdoor, and object domains. A central challenge is the severe geometric discrepancy across domains, *i.e.*, differences in scene scale, sensor noise, point density, and structural patterns, which hinders processing with a unified point-based model. To address this, we introduce a geometry-aware normalization strategy that jointly voxelizes and normalizes point clouds into a standardized geometric space, adaptively based on their geometric properties to preserve salient local structures while standardizing resolution. Built on this normalization, RAM follows a hierarchical coarse-to-fine paradigm, and employs an efficient geometric transformer that models local geometric consistency instead of globally over all points. This design mitigates discretization noise introduced by voxelization, and improves computational efficiency. Extensive experiments on 12 benchmarks demonstrate state-of-the-art performance in terms of inlier ratio and registration recall. Notably, our method outperforms the counterparts trained on each separate domain, showing superior generalization capability.

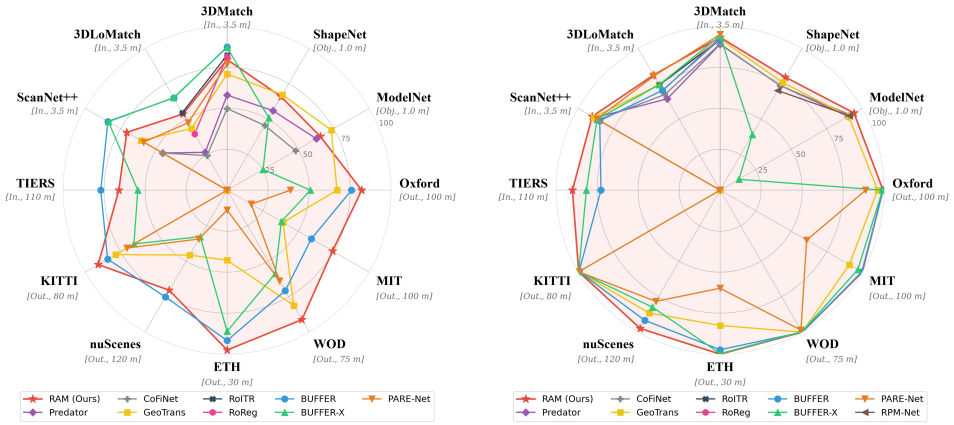
1 Introduction

Point cloud registration is a fundamental but critical task in 3D computer vision and robotics, which aims to estimate the 6-D rigid transformation aligning the source point cloud to the target. It underpins numerous applications, including 3D reconstruction, LiDAR SLAM, augmented and mixed reality (AR/XR), autonomous driving and robotic manipulation. Contemporary approaches usually follow a two-stage pipeline, which first establishes correspondences between the source and the target point clouds and then estimates the transformation based on these correspondences.

Recent learning-based methods [1–6] have made remarkable advances in point cloud registration. However, they are commonly tailored and trained for a specific *domain*. Different domains could utilize different sensors, *e.g.*, time-of-flight cameras *v.s.* LiDAR scanners, or present distinct environments, *e.g.*, indoor [7, 8], outdoor [9, 10] or object [11, 12] environments, which induces pronounced

*Corresponding authors: Qiong Li (517066092@qq.com), Kai Xu (kevin.kai.xu@gmail.com).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Inlier Ratio (b) Registration Recall

Figure 1: Register Anything Model (RAM) achieves strong generalization and robustness across a wide range of domains. Compared with the baselines, our method extracts significantly more high-quality correspondences and effectively improves the registration performance.

domain shift in the geometric characteristics. Consequently, these methods typically succeed only in domains similar to the training domain and often fail to generalize to novel domains. While recent work [5, 13] explores adapting a model pretrained on one domain to others, target-domain geometries remain unseen during training. As a result, domain shifts are only partially mitigated. In practice, this leads to inaccurate correspondences in novel domains and degraded registration accuracy.

Inspired by the recent success of vision foundation models [14–16], we consider the following question: ***can we train a point cloud registration model with strong cross-domain generalizability by leveraging diverse point cloud data spanning a wide range of domains?*** Our answer is ***yes***, but constructing such a unified model is challenging due to the irregular nature of point clouds and the substantial cross-domain geometric discrepancy, *e.g.*, scene scales, point densities, and structural patterns. At the micro-architecture level, point operators [17–20] typically encode local geometry within a fixed search radius. Due to density variability, it is suboptimal to use the same search radius across different domains. Dynamically adjusting search radius [13] requires operator kernels to model structures at diverse scales, which can complicate optimization and impair convergence. At the macro-architecture level, current registration methods [3, 20] adopt domain-specific designs, *e.g.*, downsampling factors, to better balance efficacy and efficiency: outdoor domains demand aggressive downsampling for large spatial coverage and efficiency, while indoor/object scenes benefit from finer resolutions. This heterogeneity complicates a unified architecture.

In this work, we explore cross-domain training for point cloud registration and propose a unified registration model, ***R**egister **A**nything **M**odel* (*RAM*). RAM follows a hierarchical coarse-to-fine paradigm [3, 21], which first extracts coarse-grained superpoint correspondences and then refines them into fine-grained point correspondences. We address two core questions: (1) *how to mitigate the geometric discrepancy across domains*, and (2) *how to design an architecture that remains effective and efficient across diverse conditions*. For the first problem, we introduce a geometry-aware normalization strategy that jointly voxelizes and normalizes point clouds into a domain-agnostic standardized geometric space, adaptively from their geometric properties. We cast this as a point cloud quality assessment problem: identify an as-large-as-possible voxel size that preserves salient local structures, *i.e.*, does not materially alter geometry, and normalize the resulting point clouds to a reference resolution. This allows for a unified micro architecture despite domain-dependent densities and scales. For the second problem, we further propose an efficient geometric transformer that models inter-point-cloud geometric consistency within a limited scope rather than the whole point cloud. This design alleviates the geometric noises introduced during voxelization and effectively reduces the computational cost, contributing to a better balance between feature discriminativeness and computational efficiency.

We train RAM on a collection of 4 registration datasets spanning indoor [7], outdoor [9, 10], and object [11] benchmarks, and evaluate on 12 benchmarks. Extensive experiments demonstrate strong

generalizability and robustness across diverse domains: RAM outperforms the previous state-of-the-art BUFFER-X [13] in *Registration Recall* on novel domains, and achieves an average improvement of 10.2 percentage points (pp) in *Inlier Ratio*. Our main contributions are:

- We study the efficacy of cross-domain training for point cloud registration and propose a unified registration model with strong cross-domain generalizability and robustness.
- We introduce a geometry-aware normalization strategy which jointly voxelizes and normalizes point clouds into a regularized geometry space, mitigating cross-domain geometric discrepancy.
- We devise an efficient geometric transformer which models inter-point-cloud geometric consistency within a limited scope, achieving a better balance between efficiency and efficacy.
- We conduct extensive experiments across a wide range of domains and show that our method establishes new state-of-the-art performance on 12 benchmarks.

2 Related Work

2.1 Point Cloud Registration

Point cloud registration plays a critical role in various computer vision and robotics applications. Most registration methods follow a matching-based pipeline. First, a set of correspondences between the source and the target point clouds are established by geometric feature matching. Recently, learning-based methods [1–5, 19–24] have made remarkable advances by learning discriminative geometric features in a data-driven manner. Early methods [1, 2, 25] usually first detect a set of keypoints and then match them based on their features. However, as keypoint detection is an ill-posed problem, it suffers from low repeatability and thus harms the registration performance. Recent methods [3–5, 20, 21] bypass keypoint detection by employing a hierarchical registration framework, which first extracts coarse-grained superpoint or patch correspondences and then refine them to fine-grained point correspondences. And some methods [3, 5, 19, 20, 22] design rotation-invariant or rotation-equivariant operators to improve the robustness to large rotations. After correspondences are extracted, the rigid transformation can be estimated by robust estimators such as RANSAC [26–28] and learning-based estimators [29, 30]. There are also end-to-end methods which directly regress and optimize the transformation [6, 31–33]. However, they are vulnerable to complex geometric structures and severe occlusion, which limits their application in real-world applications.

2.2 Generalizable Point Cloud Registration

Although recent learning-based methods have made remarkable progress, they usually suffer from weak generalizability across different domains due to their significant geometric discrepancy, such as scene scales, point density, noise distributions and structural patterns. A model trained on indoor datasets usually has difficulty in registering outdoor point clouds, and vice versa. Recent methods [5, 13] have studied to adapt a registration model pretrained on one domain to more domains. BUFFER-X [13] adaptively selects the voxel size based on the sphericity of point clouds and the search radius by considering the point density, which generalizes the BUFFER [5] model trained on 3DMatch to outdoor scenes. However, due to the lack of awareness of domain-specific geometries during training, it still suffers from inaccurate correspondences in novel domains, which leads to suboptimal registration in hard cases. Besides, it still needs specific tuning for different domains, *e.g.*, inlier thresholds for outlier rejection and RANSAC hyperparameters, which limits its application in real-world scenarios. In this work, we bridge this gap by constructing a unified registration model by cross-domain training, which achieves strong generalizability and robustness.

2.3 Cross-source Point Cloud Registration

Cross-source registration [34–39] is another closely related but distinct problem to cross-domain registration. Both must handle substantial variations in point density, geometric scale, sensor noise, and sampling patterns. Cross-source registration aims to align two point clouds acquired from different sources or sensing modalities, such as Kinect, LiDAR, and structure-from-motion systems. In contrast, cross-domain registration aims to develop a unified model that can register point-cloud pairs across different environments or sensors. In this work, the two point clouds within each pair are generally acquired from the same source, while the source or environment may vary across pairs. We discuss the effectiveness of our method in cross-source registration in the supplementary material.

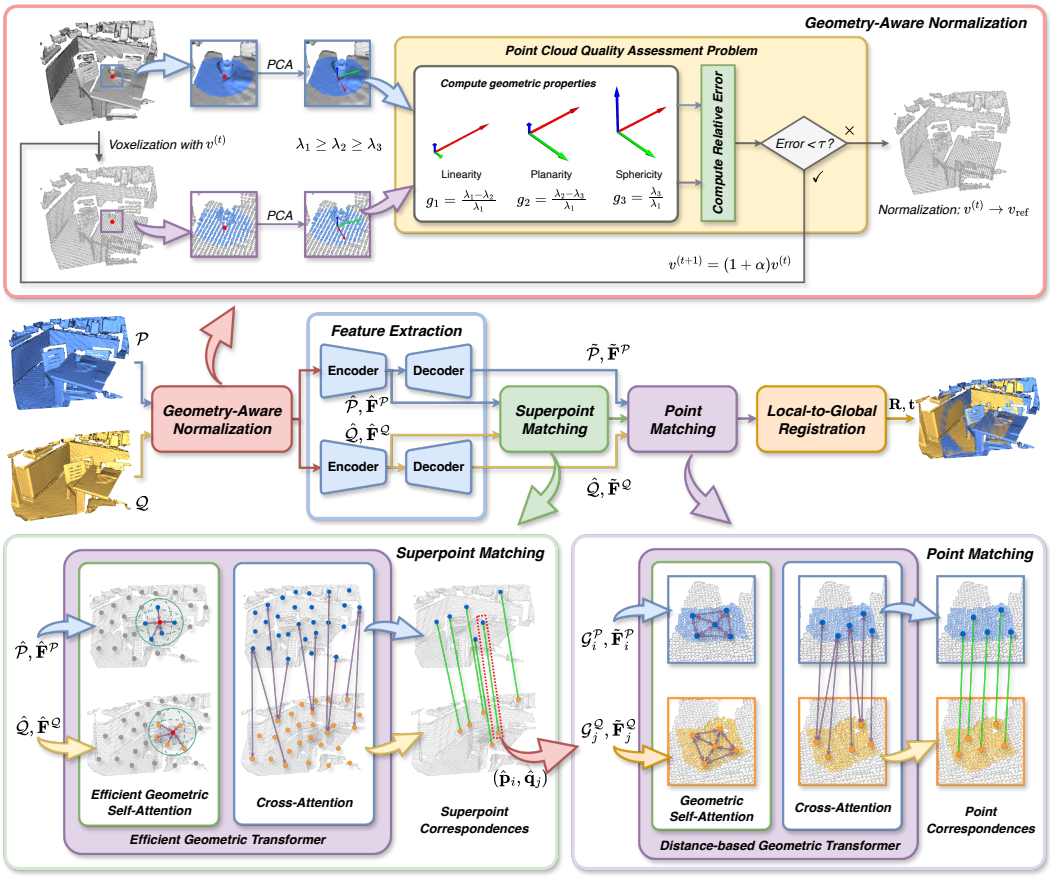


Figure 2: Overall pipeline of RAM. Point clouds are first adaptively voxelized and normalized into a standardized geometric space by a geometry-aware normalization module. Aftering extracting hierarchical features, RAM extracts superpoints correspondences with an EGeoTR, and then refines them into dense point correspondences. At last, the transformation is recovered from the correspondences.

3 Method

3.1 Overview

Given two point clouds $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \dots, M\}$ and $\mathcal{Q} = \{\mathbf{q}_j \in \mathbb{R}^3 \mid j = 1, \dots, N\}$, our goal is to estimate the rigid transformation $\mathbf{T} \in \mathcal{SE}(3)$ aligning $\mathcal{P}$ to $\mathcal{Q}$, where $\mathbf{T}$ consists of a 3D rotation $\mathbf{R} \in \mathcal{SO}(3)$ and a 3D translation $\mathbf{t} \in \mathbb{R}^3$. Our method follows the traditional registration pipeline where we first extract point correspondences $\mathcal{C} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in \mathcal{P}, \mathbf{y}_i \in \mathcal{Q}\}$, and then estimate $\mathbf{T}$ based on the correspondences by solving:

$$\mathbf{R}, \mathbf{t} = \underset{\mathbf{R} \in \mathcal{SO}(3), \mathbf{t} \in \mathbb{R}^3}{\arg\min} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{C}} \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\|^2. \tag{1}$$

Previous methods [2, 3, 5, 20, 21] assume that point clouds to register share the training domain, leading to poor generalizability across domains. In this work, we break this gap by jointly training a unified point cloud registration model on data from various domains. To address cross-domain geometric discrepancy, we first design a geometry-aware normalization module which jointly voxelizes and normalizes point clouds into a domain-agnostic standardized geometric space (Sec. 3.2). RAM then employs a hierarchical registration pipeline [3, 20, 21], which first extracts superpoint correspondences and then refine them into dense point correspondences (Sec. 3.3). Finally, the rigid transformation is estimated by LGR [3]. Fig. 2 illustrates the overall pipeline.

3.2 Geometry-aware Normalization

A unified registration model must handle the geometric discrepancy among point clouds from different domains, *e.g.*, in scene scale and point density. For instance, outdoor LiDAR scans are sparse and long-range (typically $\sim$100m), while indoor point clouds are denser and short-range (typically $<$10m). Modern point operators [1, 4, 5, 17–20, 40] encode local geometry within a fixed search radius, which is problematic across domains: a small radius on large-scale scenes yields insufficient receptive fields, whereas a large radius on small-scale scenes oversmooths local details.

To address this issue, we design a *geometry-aware normalization* module to jointly voxelize and normalize the point clouds into a domain-agnostic standardized geometric space with consistent scales and densities, enabling effective processing with a unified search radius. Our key motivation is to adaptively select a voxel size v_{raw} per point cloud pair according to their geometric properties, and then align it with a reference v_{ref} by rescaling $\mathbf{p}_{\text{norm}} = {}^{v_{\text{ref}}}/_{v_{\text{raw}}} \cdot \mathbf{p}$. We formulate the selection of v_{raw} as a *point cloud quality assessment problem*, and iteratively seek for an as-large-as-possible voxel size whose induced geometric change remains tolerable.

Given raw point clouds $\mathcal{P}$ and $\mathcal{Q}$, we initialize $v^{(0)} = \phi(\min(d(\mathcal{P}), d(\mathcal{Q})))$, where $d(\cdot)$ is the diagonal length of the bounding box, and $\phi(\cdot)$ is a monotonically increasing function (see the appendix for the definition). In t-th iteration, we voxelize $\mathcal{P}$ with $v^{(t)}$ to obtain $\mathcal{P}'$, and sample a set of seeds $\{\mathbf{s}_i\}$ from $\mathcal{P}'$ via furthest point sampling. For each seed $\mathbf{s}_i$, we collect the local patches $\mathcal{N}_i$ and $\mathcal{N}_i'$ within the radius of $r = \lambda \cdot v^{(t)}$ in $\mathcal{P}$ and $\mathcal{P}'$, respectively, where λ is a scaling factor. Inspired by [41], we compute the eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ of the covariance matrix of each patch and derive three geometric properties, *i.e.*, linearity g_1, planarity g_2 and sphericity g_3:

$$g_1 = (\lambda_1 - \lambda_2)/\lambda_1, \quad g_2 = (\lambda_2 - \lambda_3)/\lambda_1, \quad g_3 = \lambda_3/\lambda_1. \tag{2}$$

We define the geometric error e_i of $\mathbf{s}_i$ by the maximum relative error on linearity and planarity between $\mathcal{N}_i$ and $\mathcal{N}_i'$. The geometric error $e(\mathcal{P})$ between $\mathcal{P}$ and $\mathcal{P}'$ is the median geometric error over the half of seeds with the largest sphericity in $\mathcal{P}$. The same computation goes for $\mathcal{Q}$. Our insight behind this design is that oversmoothing primarily alters linearity and planarity, while sphericity serves as an indicator of local geometric complexity, filtering out geometry-less patches which are less sensitive to oversmoothing. If $e(\mathcal{P})$ and $e(\mathcal{Q})$ are both below a tolerance τ, we enlarge the voxel size by $v^{(t+1)} = (1 + \alpha)v^{(t)}$ and repeat the above process until $e \geq \tau$ or by up to T times.

3.3 Register Anything Model

Given normalized point clouds, RAM performs coarse-to-fine registration [3, 20, 21], *i.e.*, superpoint matching followed by dense point matching. For brevity, we omit the prime and reuse $\mathcal{P}$ and $\mathcal{Q}$ for the normalized point clouds in the following text.

Feature extraction. We extract hierarchical point features with a KPConv-FPN [17] backbone. Following [3], the coarsest-level points $\hat{\mathcal{P}}$ and $\hat{\mathcal{Q}}$ are used as the superpoints, while the half-resolution points $\tilde{\mathcal{P}}$ and $\tilde{\mathcal{Q}}$ support dense point matching. Their features are denoted as $\hat{\mathbf{F}}^{\mathcal{P}} \in \mathbb{R}^{\hat{M} \times d_{\text{c}}}$, $\hat{\mathbf{F}}^{\mathcal{Q}} \in \mathbb{R}^{\hat{N} \times d_{\text{c}}}$, $\tilde{\mathbf{F}}^{\mathcal{P}} \in \mathbb{R}^{\tilde{M} \times d_{\text{f}}}$ and $\tilde{\mathbf{F}}^{\mathcal{Q}} \in \mathbb{R}^{\tilde{N} \times d_{\text{f}}}$. The dense points are assigned to their nearest superpoints via point-to-node partition. The points assigned to $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{q}}_j$ are denoted as $\mathcal{G}_i^{\mathcal{P}}$ and $\mathcal{G}_j^{\mathcal{Q}}$, with gathered features $\tilde{\mathbf{F}}_i^{\mathcal{P}} \subset \tilde{\mathbf{F}}^{\mathcal{P}}$ and $\tilde{\mathbf{F}}_j^{\mathcal{Q}} \subset \tilde{\mathbf{F}}^{\mathcal{Q}}$.

Superpoint matching. Geometric transformer [3] improves superpoint feature discriminativeness by encoding inter-point-cloud global geometric consistency [3, 4, 20, 42], but their geometric structure embedding has memory complexity $O(N^2 d)$, which is prohibitive for large point clouds and typically necessitates domain-specific downsampling [3, 4, 20]. Moreover, voxelization can destabilize the geometric consistency between distant superpoints, and we observe that attention scores for such pairs are empirically low, indicating redundancy in the global geometric structure embedding.

We therefore propose an *efficient geometric transformer* (*EGeoTR*) module, which models the local geometric consistency between point clouds instead. EGeoTR interleaves local geometric self-attention encoding intra-point-cloud local geometric structures and cross-attention encoding inter-point-cloud geometric consistency to produce reliable superpoint features. Specifically, for each superpoint $\hat{\mathbf{p}}_i \in \hat{\mathcal{P}}$, let $\mathcal{N}_i$ be the indices of its k-nearest superpoints. Given input features

$\mathbf{X} \in \mathbb{R}^{\hat{M} \times d_c}$, the local geometric self-attention produces output features $\mathbf{Z} \in \mathbb{R}^{\hat{M} \times d_c}$ as:

$$\mathbf{z}_i = \sum_{j \in \mathcal{N}_i} \frac{\exp(a_{i,j})}{\sum_{k \in \mathcal{N}_i} \exp(a_{i,k})} (\mathbf{x}_j \mathbf{W}^{\mathrm{V}}), \tag{3}$$

and the attention score $a_{i,j}$ is computed as:

$$a_{i,j} = \frac{(\mathbf{x}_i \mathbf{W}^{\mathrm{Q}})(\mathbf{x}_j \mathbf{W}^{\mathrm{K}} + \mathbf{r}_{i,j} \mathbf{W}^{\mathrm{R}})^\top}{\sqrt{d_c}}, \tag{4}$$

where $\mathbf{r}_{i,j}$ is the geometric structure embedding [3], combining pair-wise distance embedding and triplet-wise angular embedding. $\mathbf{W}^{\mathrm{Q}}, \mathbf{W}^{\mathrm{K}}, \mathbf{W}^{\mathrm{V}}, \mathbf{W}^{\mathrm{R}} \in \mathbb{R}^{d_c \times d_c}$ are the learnable projection weights for queries, keys, values and embeddings, respectively. The same computation goes for $\hat{\mathcal{Q}}$. This design reduces the memory complexity of the geometric structure embedding to $O(Nkd)$, where $k \ll N$, and therefore avoids the need to alter downsampling ratios for various domains. For the cross-attention, since the geometric structure embedding is not injected, it can be easily accelerated by flash attention [43] or replaced by linear attention [44]. The resulting hybrid superpoint features are denoted as $\hat{\mathbf{H}}^{\mathcal{P}}$ and $\hat{\mathbf{H}}^{\mathcal{Q}}$.

At last, the superpoint correspondences are extracted by selecting the largest N_c entries in the correlation matrix between $\hat{\mathbf{H}}^{\mathcal{P}}$ and $\hat{\mathbf{H}}^{\mathcal{Q}}$. Following [3], we adopt dual normalization on the correlation matrix to suppress the ambiguous matches.

Point matching. Given a superpoint correspondence $(\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_j)$, we extract dense point correspondences by matching the points in $\mathcal{G}_i^{\mathcal{P}}$ and $\mathcal{G}_j^{\mathcal{Q}}$. Previous work [3, 4, 20, 45] directly match the geometric features $\tilde{\mathbf{F}}_i^{\mathcal{P}}$ and $\tilde{\mathbf{F}}_j^{\mathcal{Q}}$ from the backbone. However, within each patch, the points are so close that they have very similar features and cannot be effectively distinguished. We therefore employ a lightweight EGeoTR module comprising one local geometric self-attention and one cross-attention. Unlike the superpoint matching, we only use the pair-wise distance embedding in the local geometric self-attention for better efficiency. The resulting hybrid point features are denoted as $\tilde{\mathbf{H}}_i^{\mathcal{P}}$ and $\tilde{\mathbf{H}}_j^{\mathcal{Q}}$. Inspired by [46], we further predict the overlapping scores $\tilde{\mathbf{o}}_i^{\mathcal{P}}$ and $\tilde{\mathbf{o}}_j^{\mathcal{Q}}$ of the dense points and refines the point features as $\tilde{\mathbf{Y}}_i^{\mathcal{P}}$ and $\tilde{\mathbf{Y}}_j^{\mathcal{Q}}$ with two respective linear projections:

$$\begin{aligned} \tilde{\mathbf{o}}_i^{\mathcal{P}} = \sigma(\tilde{\mathbf{H}}_i^{\mathcal{P}} \mathbf{w}^{\mathrm{O}}),\ \tilde{\mathbf{o}}_j^{\mathcal{Q}} = \sigma(\tilde{\mathbf{H}}_j^{\mathcal{Q}} \mathbf{w}^{\mathrm{O}}), \\ \tilde{\mathbf{Y}}_i^{\mathcal{P}} = \tilde{\mathbf{H}}_i^{\mathcal{P}} \mathbf{W}^{\mathrm{F}},\ \tilde{\mathbf{Y}}_j^{\mathcal{Q}} = \tilde{\mathbf{H}}_j^{\mathcal{Q}} \mathbf{W}^{\mathrm{F}}, \end{aligned} \tag{5}$$

where $\mathbf{w}^{\mathrm{O}} \in \mathbb{R}^{d_f}$ and $\mathbf{W}^{\mathrm{F}} \in \mathbb{R}^{d_f \times d_f}$ are learnable weights, and $\sigma(\cdot)$ is the sigmoid function.

The point correspondences are extracted by jointly considering the overlapping scores and the feature similarity. Specifically, we compute the similarity matrix $\tilde{\mathbf{S}} = \tilde{\mathbf{Y}}_i^{\mathcal{P}} (\tilde{\mathbf{Y}}_j^{\mathcal{Q}})^\top$ between $\mathcal{G}_i^{\mathcal{P}}$ and $\mathcal{G}_j^{\mathcal{Q}}$, and inject the overlapping scores to obtain the confidence matrix $\mathbf{S}$:

$$\mathbf{S} = \left[\tilde{\mathbf{o}}_i^{\mathcal{P}} (\tilde{\mathbf{o}}_j^{\mathcal{Q}})^\top\right] \odot \varphi_{\text{row}}(\tilde{\mathbf{S}}) \odot \varphi_{\text{col}}(\tilde{\mathbf{S}}), \tag{6}$$

where $\odot$ denotes the Hadamard product and $\varphi_{\text{row}}(\cdot)$ and $\varphi_{\text{col}}(\cdot)$ are the row-wise and column-wise softmax operations. The local point correspondences are extracted by mutual top-k selection on $\mathbf{S}$, and the global point correspondences aggregate the local correspondences from all superpoint correspondences. The final transformation is estimated by LGR [3] from the point correspondences.

3.4 Loss Functions

RAM is trained with three loss functions: (1) an overlap-aware circle loss [3] $\mathcal{L}_{\text{oc}}$ to supervise the superpoint features, (2) a negative log-likelihood loss $\mathcal{L}_{\text{nll}}$ to supervise the confidence matrices, and (3) a binary cross-entropy loss $\mathcal{L}_{\text{bce}}$ to supervise the overlapping scores. The overall loss is $\mathcal{L} = \lambda_{\text{oc}} \mathcal{L}_{\text{oc}} + \lambda_{\text{nll}} \mathcal{L}_{\text{nll}} + \lambda_{\text{bce}} \mathcal{L}_{\text{bce}}$.

4 Experiments

4.1 Benchmarks and metrics.

Benchmarks. Following [13], we evaluate RAM on 12 point cloud registration benchmarks, including 4 indoor benchmarks: 3DMatch [7], 3DLoMatch [2], ScanNet++ [47] and TIERS [48], 6

Table 1: Results on indoor benchmarks. **Boldfaced** numbers highlight the best and the second best are underlined. † and ‡ denote the models pretrained on 3DMatch and KITTI, respectively. $^{\triangledown}$ denotes the models without outlier rejection. * denotes the model trained on indoor data. "N/A" indicates the model fails to achieve reasonable registrations.

Model	3DMatch		3DLoMatch		ScanNet++		TIERS	
	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)
Predator [2]	58.0	89.0	26.7	64.4	45.4	88.3	N/A	
CoFiNet [21]	49.8	89.3	24.4	67.4	45.4	85.8	N/A	
GeoTrans [3]	70.9	92.4	43.5	74.3	60.6	88.9	N/A	
RoITR [4]	82.4	91.9	54.6	74.1	-	-	-	-
BUFFER [5]	**87.4**	92.1	65.0	70.3	**83.9**	84.7	**77.0**	72.6
RoReg [54]	81.6	93.0	39.6	70.1	-	-	-	-
PARENet [20]	76.9	**95.0**	47.5	80.5	58.7	86.8	N/A	
BUFFER-X† [13]	87.1	94.2	**65.2**	74.3	83.5	86.6	54.4	81.5
BUFFER-X‡ [13]	83.1	92.3	59.5	67.1	79.2	86.6	60.8	84.6
BUFFER$^{\triangledown}$ [5]	50.2	89.6	21.1	60.9	36.5	64.5	44.1	69.1
BUFFER-X$^{\triangledown\dagger}$ [13]	35.1	90.2	14.0	56.7	26.7	83.3	31.7	81.9
RAM* (*ours*)	75.0	92.4	47.7	77.8	65.9	89.2	48.5	81.3
RAM (*ours*)	79.7	93.3	53.7	**81.1**	70.6	**90.0**	66.0	**90.0**

outdoor benchmarks: KITTI [9], nuScenes [10], ETH [49], MIT [50], WOD [51, 52] and Oxford [53], and 2 object benchmarks: ModelNet [11] and ShapeNet [12]. We train RAM on a hybrid set combining 3DMatch, KITTI, nuScenes and ModelNet. Additional details are in the appendix.

Metrics. Following [3, 13], we report the results on 4 metrics: (1) *Inlier Ratio* (IR), the percentage of correspondences whose residuals are below a certain threshold under the ground-truth transformation. (2) *Relative Rotation Error* (RRE), the geodesic distance between the estimated and the ground-truth rotations. (3) *Relative Translation Error* (RTE), the ℓ_2-distance between the estimated and the ground-truth translations. (4) *Registration Recall* (RR), the percentage of successfully registered pairs. A pair is successful if its RMSE is below 0.2m (3DMatch/3DLoMatch) or if both RRE and RTE are below certain thresholds (other benchmarks). We adopt stricter RRE and RTE thresholds to enable more rigorous performance assessment. Please refer to the appendix for more details.

4.2 Experiments on Indoor Benchmarks

Baselines. We compare against 9 baselines: Predator [2], CoFiNet [21], GeoTrans [3], RoITR [4], BUFFER [5], RoReg [54], PEAL [42], PARENet [20] and BUFFER-X [13]. The first 8 baselines are trained on 3DMatch. And we compare with two BUFFER-X variants based on BUFFER pretrained on 3DMatch (marked by †) and KITTI (marked by ‡), respectively.

Quantitative results on 3DMatch and 3DLoMatch. Tab. 1 (left) shows the results on 3DMatch and 3DLoMatch. For *Inlier Ratio*, RAM is better than or on par with most baselines except BUFFER. This is because BUFFER integrates a learnable outlier rejection module, which significantly improves the inlier ratio. For *Registration Recall*, RAM outperforms all baselines on the more challenging 3DLoMatch, demonstrating the strong robustness of our method to severe occlusion. And RAM trained on hybrid data also surpasses the model trained on in-domain data (marked by *). These results indicate that exposure to diverse domains enriches geometric patterns during training, which effectively enhances the learned point features and yields more accurate correspondences. This is consistent with our goal to construct a unified registration model via cross-domain training.

Quantitative results on unseen benchmarks. We further evaluate the generalizability on two unseen indoor benchmarks, ScanNet++ and TIERS. RAM surpasses BUFFER-X respectively by 3.4 and 5.4 percentage points (pp) in *Registration Recall* on ScanNet++ and TIERS, respectively, and by 5.2 pp in *Inlier Ratio* on TIERS. These results have demonstrated the superior generalizability of our method to unseen domains. Note that ScanNet++ has a similar data distribution with 3DMatch, so BUFFER-X based on 3DMatch outperforms the variant based on KITTI. While TIERS is captured with a spinning LiDAR scanner and thus resembles an outdoor domain, favouring the KITTI-based variant. On the contrary, our method consistently outperforms both variants of BUFFER-X, further proving the efficacy of cross-domain training for improving the registration generalizability.

Table 2: Results on outdoor benchmarks. **Boldfaced** numbers highlight the best and the second best are underlined. † and ‡ denote the models pretrained on 3DMatch and KITTI, respectively. * denotes the model trained on outdoor data. $^{\diamond}$ denotes the model trained without nuScenes.

Model	KITTI		nuScenes		ETH		WOD		MIT		Oxford	
	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)
GeoTrans [3]	78.5	**99.5**	45.7	86.5	42.7	82.5	81.2	**100.0**	39.3	90.9	67.0	96.0
BUFFER [5]	84.1	99.3	**75.2**	91.7	91.7	97.3	70.8	**100.0**	59.3	99.7	75.8	**99.3**
PARENet [20]	70.5	**99.5**	34.4	78.3	12.1	59.8	64.0	98.5	17.0	60.9	38.6	88.7
BUFFER-X† [13]	65.6	**99.5**	32.7	82.4	86.1	99.7	58.7	99.7	38.2	96.9	50.8	99.0
BUFFER-X‡ [13]	73.9	**99.5**	52.6	93.5	90.8	**100.0**	65.2	99.7	51.6	98.7	59.4	**99.3**
RAM* (*ours*)	89.5	**99.5**	70.0	96.8	93.8	97.9	90.4	**100.0**	72.3	98.7	79.2	99.0
RAM$^{\diamond}$ (*ours*)	90.5	**99.5**	59.4	95.2	96.7	99.3	90.0	**100.0**	62.5	97.4	80.6	**99.3**
RAM (*ours*)	**90.7**	**99.5**	70.5	**97.4**	**97.5**	**100.0**	**91.2**	**100.0**	**74.2**	**100.0**	**82.0**	**99.3**

Comparison with BUFFER and BUFFER-X without outlier rejection. We additionally report the performance of BUFFER and BUFFER-X before the outlier-rejection module (marked by $^{\triangledown}$). The dedicated outlier-rejection stage substantially improves their performance. Before outlier rejection, RAM consistenctly outperforms both baselines across all benchmarks, indicating that our method produces more robust initial correspondences. After outlier rejection, RAM still achieves higher *Registration Recall* on 3DLoMatch, ScanNet++, and TIERS, demonstrating stronger generalization to challenging unseen indoor benchmarks.

4.3 Experiments on Outdoor Benchmarks

Quantitative results on KITTI and nuScenes. We first compare against 3 baselines, *i.e.*, GeoTrans [3], BUFFER [5], and PARENet [20], all trained on KITTI. As shown in Tab. 2 (top), RAM outperforms all baselines on both benchmarks, *e.g.*, by 6.6 pp in *Inlier Ratio* on KITTI, indicating that cross-domain training yields more discriminative features and generates more accurate correspondences. On nuScenes, despite a lower *Inlier Ratio*, our method achieves substantially higher *Registration Recall*, demonstrating stronger robustness to difficult cases. Additionally, we report results for RAM trained without nuScenes (marked by $^{\diamond}$). This variant achieves higher *Registration Recall* than the baselines on most benchmarks, except MIT, further validating our method. Because both nuScenes and MIT employ LiDAR sensors with fewer beams than KITTI, omitting nuScenes during training slightly degrades performance relative to the model trained with nuScenes.

Quantitative results on unseen benchmarks. We further assess the generalizability on 4 unseen outdoor benchmarks, *i.e.*, ETH, WOD, MIT and Oxford. As these benchmarks all adopts spinning LiDAR scanners, most baselines except PARENet generalize reasonably well. RAM achieves the best performance on all metrics across all benchmarks, which again underscoring the advantage of cross-domain training for robust generalizability. Note that the outdoor-only RAM (marked by *) trained on KITTI and nuScenes outperforms the hybrid model without nuScenes on most benchmarks, suggesting that domain-relevant training data remain important for performance within each domain.

4.4 Experiments on Object Benchmarks

We compare with 5 baselines: RPM-Net [6], Predator [2], CoFiNet [21], GeoTrans [3], and BUFFER-X [5], where the first 4 baselines are trained on ModelNet, while BUFFER-X uses the weights pretrained on 3DMatch and KITTI. As shown in Tab. 3, our method consistently outperforms all object baselines by a large margin on both ModelNet and ShapeNet. We notice that GeoTrans attains better results in *Inlier Ratio*. This is because object point clouds are usually small, so the global geometric structure embedding is less affected by the voxelization noises and contributes to better feature representation in most cases. However, thanks to the rich geometric patterns from diverse domains, RAM extracts more accurate correspondences in the harder cases, thus contributing to better *Registration Recall*. Another important observation is that both variants of BUFFER-X fail to achieve reasonable results on the object registration benchmarks, which unveils the significant difficulty of generalization from scene registration to object registration due to the severe geometric discrepancy. In contrast, benefiting from the diverse data from various domains during training, our method achieves strong generalizability across a wide range of domains.

Table 3: Results on object benchmarks. **Boldfaced** numbers highlight the best and the second best are underlined. † and ‡ denote the models pretrained on 3DMatch and KITTI, respectively. ⋆ denotes the model trained on object data.

Model	ModelNet				ShapeNet			
	IR(%)	RR(%)	RRE(°)	RTE(m)	IR(%)	RR(%)	RRE(°)	RTE(m)
RPM-Net [6]	-	90.9	2.412	0.030	-	70.2	6.177	0.058
Predator [2]	62.8	90.1	2.614	0.033	56.0	73.6	5.668	0.067
CoFiNet [21]	48.2	91.8	2.802	0.038	45.7	73.5	6.834	0.068
GeoTrans [3]	**73.3**	90.4	2.636	0.032	**67.3**	76.0	**4.492**	0.053
BUFFER-X† [13]	25.3	13.3	95.483	0.608	50.9	39.3	67.315	0.396
BUFFER-X‡ [13]	46.4	34.9	62.221	0.401	65.0	55.3	47.209	0.276
RAM⋆ (*ours*)	66.4	90.9	2.026	0.026	65.6	76.0	4.684	0.053
RAM (*ours*)	65.6	**94.2**	**1.875**	**0.023**	65.9	**79.6**	4.592	**0.047**

Table 4: Ablation studies. **Boldfaced** numbers highlight the best and the second best are underlined. **OS**: overlap scores. **O&T**: overlap scores and transformer. **GN**: geometry-aware normalization.

Model	ScanNet++		TIERS		ETH		WOD		MIT		Oxford		ShapeNet	
	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)	IR(%)	RR(%)
(a.1) hybrid⋆	**70.6**	**90.0**	**66.0**	**90.0**	**97.5**	**100.0**	**91.2**	**100.0**	**74.2**	**100.0**	**82.0**	**99.3**	**65.9**	**79.6**
(a.2) indoor	65.9	89.2	48.5	81.3	77.9	87.3	66.2	98.5	23.8	58.3	64.4	96.7	21.0	30.7
(a.3) outdoor	41.1	72.5	62.3	87.5	93.8	97.9	90.4	**100.0**	72.3	98.7	79.2	99.0	21.0	28.8
(a.4) object	28.9	56.0	42.7	68.4	32.3	39.5	36.2	66.2	31.3	75.2	52.0	86.7	65.6	76.0
(b.1) w/ OS⋆	**70.6**	**90.0**	**66.0**	**90.0**	**97.5**	**100.0**	**91.2**	**100.0**	**74.2**	**100.0**	**82.0**	**99.3**	**65.9**	79.6
(b.2) w/o OS	68.3	88.9	64.7	89.1	96.5	99.4	90.8	**100.0**	72.9	99.1	80.9	**99.3**	63.6	**80.3**
(b.3) w/o O&T	63.7	88.9	61.0	88.7	96.3	99.0	86.0	**100.0**	69.3	99.1	74.9	99.0	58.6	79.6
(c.1) GN⋆	**70.6**	90.0	**66.0**	**90.0**	**97.5**	**100.0**	91.2	**100.0**	**74.2**	**100.0**	82.0	**99.3**	**65.9**	**79.6**
(c.2) fixed	66.8	**90.6**	47.8	89.2	62.4	99.0	**91.3**	**100.0**	72.0	99.1	**86.0**	**99.3**	65.9	79.6

4.5 Ablation Studies

At last, we conduct extensive ablation studies to better understand the design choices of RAM on the 7 unseen benchmarks, *i.e.*, ScanNet++, TIERS, ETH, WOD, MIT, Oxford and ShapeNet.

Cross-domain training. We first investigate how cross-domain training affects the generalizability, as shown in Tab. 4 (a). We compare 4 variants of RAM, respectively trained on (1) hybrid data, (2) indoor data, (3) outdoor data, and (4) object data. For *Registration Recall*, the indoor model performs quite well on most outdoor benchmarks except MIT. The outdoor model performs well on TIERS but fails to generalize to ScanNet++. Both models fail on the object benchmark, while the object model also fails on the scene-level benchmarks. For *Inlier Ratio*, all models suffer from significant performance drop on different domains. The hybrid model achieves the best performance on all benchmarks, demonstrating the efficacy of cross-domain training.

Point matching. We then study the efficacy of the designs in the point matching stage in Tab. 4 (b). We compare 3 variants of RAM: (1) the full model, (2) the model without the overlap scores, and (3) the model further without the geometric transformer. By predicting the overlapping scores, the model suppresses the non-overlapping points during matching and thus yields more accurate correspondences. And removing the transformer causes a significant performance degradation, further confirming the importance of the inter-point-cloud information exchange in point matching.

Geometry-aware normalization. We evaluate the efficacy of our geometry-aware normalization by comparing with the model with static voxel sizes in Tab. 4 (c). Following [3, 5], we set the voxel size to 2.5cm for indoor and object benchmarks, and 30cm for outdoor benchmarks. Both models use the same weights trained on static voxel sizes. The model with geometry-aware normalization consistently outperforms the static counterpart on most benchmarks. Note that the benefit of our design is not limited to the average improvement. It eliminates the need to manually select a domain-specific voxel size at inference time. This is particularly important in practical applications where an agent may frequently transition between environments with substantially different geometric scales.

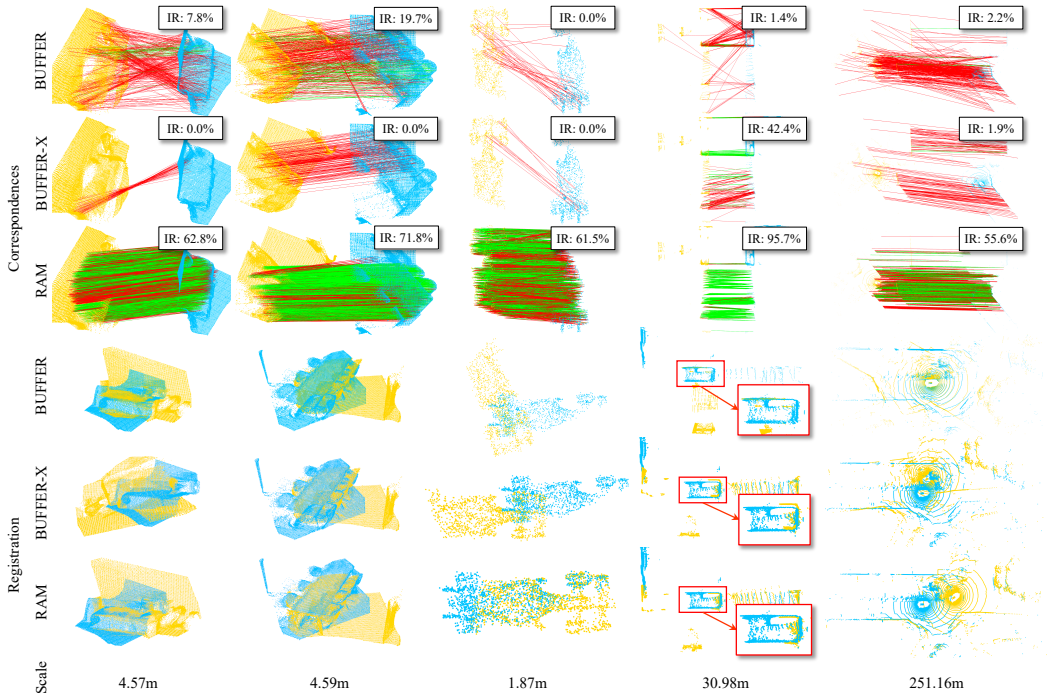


Figure 3: Visualization of the registration results on various benchmarks.

4.6 Qualitative Results

Fig. 3 visualizes some registration results of BUFFER, BUFFER-X and RAM on various benchmarks. Our method can effectively extract accurate correspondences across different domains, despite significant geometric discrepancy in terms of scene scales, point densities, structural patterns. And our method is still robust in case of incomplete geometry, duplicated objects, and symmetric structures.

5 Conclusion

In this paper, we investigated cross-domain training for point cloud registration and introduced a unified registration model, RAM. Extensive experiments across a broad set of benchmarks demonstrate that RAM delivers strong cross-domain generalizability and robustness, achieving state-of-the-art performance in diverse indoor, outdoor, and object registration scenarios.

Despite these advances, RAM can be improved in several respects. First, the KPConv-based backbone is still rotation-variant. Although RAM already shows promising robustness to large rotations as demonstrated in the appendix, it may still struggle under extreme scenarios. We assessed recent rotation-equivariant backbones, but their memory footprint currently prevents processing large-scale point clouds within a unified architecture. Second, RAM may not scale to extremely large point clouds, *e.g.*, urban-scale UAV captures. A practical remedy is to partition scenes into manageable tiles and process them separately, potentially with overlap and streaming. Third, we observe that distant outliers in point clouds can inflate the initial voxel size and oversmooth geometry. This could be mitigated by integrating a separate point cloud outlier removal step.

Beyond registration, RAM can serve as a general-purpose model for point cloud processing, and we hope it will help catalyze future research toward point cloud foundation models.

6 Acknowledgements

This work is supported in part by the NSFC (62402516, 62325211, 62132021, 62402505) and the Research Fund of Jiangsu Key Laboratory of AI for Industries (E6420016G8).

References

[1] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *CVPR*, pages 8958–8966, 2019.

[2] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, pages 4267–4276, 2021.

[3] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, pages 11143–11152, 2022.

[4] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *CVPR*, pages 5384–5393, 2023.

[5] Sheng Ao, Qingyong Hu, Hanyun Wang, Kai Xu, and Yulan Guo. Buffer: Balancing accuracy, efficiency, and generalizability in point cloud registration. In *CVPR*, pages 1255–1264, 2023.

[6] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *CVPR*, pages 11824–11833, 2020.

[7] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, pages 1802–1811, 2017.

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012.

[10] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020.

[11] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015.

[12] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[13] Minkyun Seo, Hyungtae Lim, Kanghee Lee, Luca Carlone, and Jaesik Park. Buffer-x: Towards zero-shot point cloud registration in diverse scenes. In *ICCV*, pages 3851–3862, 2025.

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023.

[15] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[16] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 37:21875–21911, 2024.

[17] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, pages 6411–6420, 2019.

[18] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019.

[19] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *CVPR*, pages 11753–11762, 2021.

[20] Runzhao Yao, Shaoyi Du, Wenting Cui, Canhui Tang, and Chengwu Yang. Pare-net: Position-aware rotation-equivariant networks for robust point cloud registration. In *ECCV*, pages 287–303. Springer, 2024.

[21] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *NeurIPS*, 34:23872–23884, 2021.

[22] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *ACM MM*, pages 1630–1641, 2022.

[23] Xiaoshui Huang, Wentao Qu, Yifan Zuo, Yuming Fang, and Xiaowei Zhao. Imfnet: Interpretable multimodal fusion for point cloud registration. *RAL*, 7(4):12323–12330, 2022.

[24] Xiaoshui Huang, Zhou Huang, Yifan Zuo, Yongshun Gong, Chengdong Zhang, Deyang Liu, and Yuming Fang. Psreg: Prior-guided sparse mixture of experts for point cloud registration. In *AAAI*, volume 39, pages 3788–3796, 2025.

[25] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *CVPR*, pages 6359–6367, 2020.

[26] 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.

[27] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. In *CVPR*, pages 13221–13231, 2022.

[28] Xiyu Zhang, Jiaqi Yang, Shikun Zhang, and Yanning Zhang. 3d registration with maximal cliques. In *CVPR*, pages 17745–17754, 2023.

[29] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, pages 2514–2523, 2020.

[30] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *CVPR*, pages 15859–15869, 2021.

[31] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *CVPR*, pages 7163–7172, 2019.

[32] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *CVPR*, pages 8893–8902, 2021.

[33] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *ICCV*, pages 3132–3141, 2021.

[34] Xiaoshui Huang, Jian Zhang, Lixin Fan, Qiang Wu, and Chun Yuan. A systematic approach for cross-source point cloud registration by preserving macro and micro structures. *IEEE TIP*, 26(7):3261–3276, 2017.

[35] Xiaoshui Huang, Jian Zhang, Qiang Wu, Lixin Fan, and Chun Yuan. A coarse-to-fine algorithm for matching and registration in 3d cross-source point clouds. *IEEE TCSVT*, 28(10):2965–2977, 2017.

[36] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Cross-source point cloud registration: Challenges, progress and prospects. *Neurocomputing*, 548:126383, 2023.

[37] X Huang, G Mei, J Zhang, and R Abbas. A comprehensive survey on point cloud registration. arxiv 2021. *arXiv preprint arXiv:2103.02690*, 2021.

[38] Guiyu Zhao, Zewen Du, Zhentao Guo, and Hongbin Ma. Vrhcf: Cross-source point cloud registration via voxel representation and hierarchical correspondence filtering. In *ICME*, pages 1–6. IEEE, 2024.

[39] Guiyu Zhao, Zhentao Guo, Zewen Du, and Hongbin Ma. Cross-pcr: A robust cross-source point cloud registration framework. In *AAAI*, volume 39, pages 10403–10411, 2025.

[40] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *CVPR*, pages 3173–3182, 2021.

[41] Evangelos Alexiou, Xuemei Zhou, Irene Viola, and Pablo Cesar. Pointpca: Point cloud objective quality assessment using pca-based descriptors. *arXiv preprint arXiv:2111.12663*, 2021.

[42] Junle Yu, Luwei Ren, Yu Zhang, Wenhui Zhou, Lili Lin, and Guojun Dai. Peal: Prior-embedded explicit attention learning for low-overlap point cloud registration. In *CVPR*, pages 17702–17711, 2023.

[43] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *NeurIPS*, 37:68658–68685, 2024.

[44] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[45] Fan Yang, Lin Guo, Zhi Chen, and Wenbing Tao. One-inlier is first: Towards efficient position encoding for point cloud registration. *NeurIPS*, 35:6982–6995, 2022.

[46] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *ICCV*, pages 17627–17638, 2023.

[47] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, pages 12–22, 2023.

[48] Qingqing Li, Xianjia Yu, Jorge Pena Queralta, and Tomi Westerlund. Multi-modal lidar dataset for benchmarking general-purpose localization and mapping algorithms. In *IROS*, pages 3837–3844. IEEE, 2022.

[49] 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.

[50] Yulun Tian, Yun Chang, Long Quang, Arthur Schang, Carlos Nieto-Granda, Jonathan P How, and Luca Carlone. Resilient and distributed multi-robot visual slam: Datasets, experiments, and lessons learned. In *IROS*, pages 11027–11034. IEEE, 2023.

[51] 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 *CVPR*, pages 2446–2454, 2020.

[52] 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 *CVPR*, pages 20816–20826, 2024.

[53] 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 *IROS*, pages 4353–4360. IEEE, 2020.

[54] Haiping Wang, Yuan Liu, Qingyong Hu, Bing Wang, Jianguo Chen, Zhen Dong, Yulan Guo, Wenping Wang, and Bisheng Yang. Roreg: Pairwise point cloud registration with oriented descriptors and local rotations. *IEEE TPAMI*, 45(8):10376–10393, 2023.

A Appendix

A.1 Implementation Details

Voxel size initialization. We set the reference voxel size $v_{\text{ref}} = 0.025\text{m}$, a conventional choice for 3DMatch. And we define $\phi(\cdot)$ as a monotonically increasing piecewise function:

$$\phi(x) = \begin{cases} 0.1 \log_2 \left(\frac{x}{25}\right), & x \geq 50 \\ 0.001\,(x-25) + 0.075, & 25 \leq x < 50 \\ 0.0025\,(x-1) + 0.015, & x < 25 \end{cases} . \tag{7}$$

This function is designed based on several empirical anchor points, *i.e.*, $\phi(1) = 0.015\text{m}$, $\phi(5) = 0.025\text{m}$, $\phi(50) = 0.1\text{m}$ and $\phi(200) = 0.3\text{m}$. For point clouds smaller than 25m, the environments range from rooms to halls or buildings, so we want the initial voxel size to increase rapidly to reflect this change. For point clouds larger than 50m, which typically correspond to outdoor environments, an excessively large voxel size can impair registration accuracy; therefore, we employ a logarithmic function to prevent the voxel size from growing too quickly. We sample 512 seeds for each point cloud, and collect the neighbors within $8v^{(t)}$ for each seed to compute the geometric properties. Moreover, we set the relative error threshold $\tau = 0.15$ and the scaling factor $\alpha = 0.05$. The normalization process is iterated for up to $T = 5$ times.

Model details. We use a 4-stage KPConv-FPN backbone, with the initial voxel size of 0.025m. The point cloud resolution is halved after each stage, while the feature channels are doubled. The

Table 5: Metric thresholds of all benchmarks.

	IR(m)	RRE(°)	RTE(m)	RE(m)
3DMatch [7]	0.1	-	-	0.2
3DLoMatch [2, 7]	0.1	-	-	0.2
ScanNet++ [47]	0.1	5.0	0.3	-
TIERS [48]	0.8	5.0	0.6	-
KITTI [9]	1.0	5.0	0.6	-
nuScenes [10]	1.0	5.0	0.6	-
ETH [49]	0.8	2.0	0.3	-
WOD [51]	1.0	5.0	0.6	-
MIT [50]	1.0	5.0	0.6	-
Oxford [53]	1.0	5.0	0.6	-
ModelNet [11]	0.1	5.0	0.1	-
ShapeNet [12]	0.1	5.0	0.1	-

search radius of KPConv is $2.5\times$ of the voxel size of each stage. For EGeoTR, we use 128 neighbors for each superpoint to compute the geometric structure embedding. For other hyperparameters, we follow the 3DMatch settings in GeoTrans [3].

Training and testing. Our model is implemented and trained with an Intel i7-12700K CPU and an NVIDIA RTX 5090 GPU. We use AdamW optimizer to train all models for 3M iterations, with the batch size of 1 and the weight decay of 10^{-4}. We warm up the training for 25K iterations, with the learning rate linearly increasing from 10^{-5} to 10^{-4}. The learning rate then decreases to 10^{-5} with cosine annealing. We checkpoint the model every 50K iterations and use the averaged model of the last 30 checkpoints for evaluation. Our model is trained on the hybrid data from 3DMatch, KITTI, nuScenes and ModelNet. For simplicity, we do not balance the samples from each dataset when constructing the training data. The same augmentations as [3] are used in the experiments. Following [20], we employ the random crop augmentation on 3DMatch. And we use the same augmentations as KITTI on nuScenes. We do not use geometry-aware normalization during training for efficiency, and set the voxel size to 0.025m for 3DMatch/ModelNet, and 0.3m for KITTI and nuScenes. To compensate, we adopt extra random scaling augmentation ranging from $[0.8, 1.2]$.

A.2 Benchmarks

3DMatch. 3DMatch [7] is a widely used indoor point cloud registration dataset, which consists of 62 scenes captured by RGB-D sensors. We follow the standard split of 46 scenes for training, 8 for validation and 8 for testing. We use the same approach as [2] to generate point cloud pairs. During testing, the overlap ratios of the point cloud pairs in 3DMatch are above 30%, and those in 3DLoMatch are between 10% and 30%. There are 20641 pairs for training, 1331 for validation, 1623 and 1781 for testing on 3DMatch and 3DLoMatch, respectively.

ScanNet++. ScanNet++ [47] is a large-scale 3D reconstruction dataset. We use the depth images captured using an iPhone 13 Pro. We use the same approach as [13] to process the dataset, resulting in 2050 point cloud pairs for testing.

TIERS. TIERS [48] is multi-modal SLAM dataset captured by various types of LiDAR scanners. Following [13], we use the point clouds captured by Velodyne VLP-16, Ouster OS1-64, and Ouster OS0-128 from the sequences `Indoor06-Indoor11` (except `Indoor07`) sequences. We select the point cloud pairs which are at most 2m apart, obtaining 870 testing pairs.

KITTI. KITTI [9] is an autonomous driving dataset collected by a 64-beam LiDAR sensor. Following [2, 3, 13], we use sequences 0-5 for training, 6-7 for validation and 8-10 for testing. We use the point cloud pairs which are at most 10m apart and refine their poses with ICP following previous work. There are 1358 training pairs, 180 validation pairs, and 555 testing pairs, respectively.

nuScenes. nuScenes [10] is also an autonomous driving dataset captured by a 32-beam LiDAR scanner. We use the same method as KITTI to generate the point cloud pairs, and obtain 20587/1271/1330 pairs for training, validation and testing, respectively.

ETH. ETH [49] contains 4 outdoor scenes captured by a Hokuyo UTM-30LX LiDAR scanner. We follow the settings in [5, 13], which contains 713 point cloud pairs for testing.

Table 6: Computational time on 3DMatch. All results are measured in seconds.

Model	Geometry-aware Normalization	Data Preparation	Model Inference	Total
GeoTrans [3]	-	0.246	0.062	0.308
BUFFER [5]	-	0.173	0.148	0.321
BUFFER-X [13]	-	0.133	0.752	0.885
PARENet [20]	-	0.234	0.075	0.309
RAM (*ours*)	0.212	0.258	0.055	0.525

Table 7: Detailed computational time of RAM on 3DMatch. All results are measured in seconds.

Geometry-aware Normalization	Data Preparation	Feature Extraction	Superpoint Matching	Point Matching	Local-to-Global Registration
0.212	0.258	0.014	0.022	0.010	0.009

WOD. Waymo Open Dataset [51] is a large-scale autonomous driving dataset published by Waymo. We follow the settings in [52] to select the point cloud pairs which are at most 10m apart. At a result, there are 130 testing pairs.

MIT. MIT comes from the `10_14_acl_jackal` sequence of the Kimera-Multi dataset [50], which present a outdoor campus environment. The point clouds are captured by a Velodyne VLP 16 sensor, which are relatively sparse. Following [13], we use the point cloud pairs which are at most 5m apart, resulting in 230 pairs for testing.

Oxford. Oxford is extracted from the Newer College dataset [53], which consists of the `01`, `05`, and `07` sequences. We follow [13] to generate 301 testing point cloud pairs by choosing those which are at most 5m apart.

ModelNet. ModelNet contains man-made CAD models from 40 categories. We use the first 20 categories for training/validation, and the rest 20 for testing. The 8 symmetric categories are ignored. We follow the data augmentation in [3] to generate the point cloud pairs. As a result, we have 4194 CAD models for training, 1002 for validation, and 1146 for testing.

ShapeNet. ShapeNet is a large-scale CAD model dataset organized according to the WordNet taxonomy. We use the ShapeNetCore subset, which contains 55 categories of common objects. We randomly sample 50 CAD models from each categories, obtaining 2750 models for testing. For each model, we uniformly sample 8192 points to produce the point cloud. And we employ the same data augmentation as ModelNet to generate the testing pairs.

A.3 Metrics

Following common practice [2, 3, 13, 20], we use different methods to compute the *Registration Recall* for different benchmarks. For 3DMatch and 3DLoMatch, a point cloud pair is considered successfully registered if its registration error is below 0.2m. For other benchmarks, a registration is successful if its relative rotation error is below τ_{rre} and its relative translation error is below τ_{rte}. We use stricter thresholds than BUFFER-X [13] to better evaluate the performance of different methods. The threshold values for each benchmark are shown in Tab. 5.

A.4 Additaional Results

Computational efficiency. We compare the computational time for GeoTrans [3], BUFFER [5], BUFFER-X [13], PARENet [20], and RAM in Tab. 6, averaged over all test cases in 3DMatch. RAM has the fastest model-inference time among the compared methods, supporting the computational advantage of EGeoTR. And the detailed computational time of RAM is shown in Tab. 7. The first two stages are executed on the CPU, while the remaining stages run on the GPU. Geometry-aware normalization takes a comparable amount of time to data preparation (*i.e.*, building hierarchical computation graphs for point clouds). In particular, we observe that most of the runtime of geometry-aware normalization is spent on constructing KD-trees and querying neighbors on the raw point clouds. In our current implementation, the KD-trees for the raw point clouds are rebuilt at every

Table 8: Results on 3DMatch and 3DLoMatch with extended training samples. **Boldfaced** numbers highlight the best.

Data	# Epochs	3DMatch		3DLoMatch	
		IR(%)	RR(%)	IR(%)	RR(%)
indoor	60	75.0	92.4	47.7	77.8
indoor	120	76.9	92.8	50.3	79.2
hybrid	60	**79.7**	**93.3**	**53.7**	**81.1**

Table 9: Results on 3DMatch with large rotations.

Model	3DMatch		3DLoMatch	
	IR(%)	RR(%)	IR(%)	RR(%)
PARENet [20] (standard)	73.6	97.0	44.3	80.5
PARENet [20] (rotated)	73.7 (0.1↑)	96.7 (0.3↓)	44.8 (0.5↑)	81.1 (0.6↑)
RAM (*ours*, standard)	79.7	96.2	53.7	82.4
RAM (*ours*, rotated)	78.9 (0.8↓)	95.5 (0.7↓)	52.7 (1.0↓)	79.7 (2.7↓)

iteration, Caching or reusing the neighborhood structures should reduce this overhead. This stage can also be accelerated by pre-voxelization with a small voxel size (*e.g.*, 0.01m). In practice, all stages can be pipelined, yielding an inference speed of around 4 fps.

Volume of training data. As the hybrid data contains more training samples (approximately 20K pairs from 3DMatch, 1.3K pairs from KITTI, 20K pairs from nuScenes, and 4K pairs from ModelNet), we then extend the training of the indoor-only RAM to 120 epochs, which results in approximately the same total number of sample exposures as the hybrid model while retaining the same architecture and optimization settings. As shown in Tab. 8, even after matching the total training exposure, the hybrid model consistently outperforms the indoor-only model on both 3DMatch and 3DLoMatch. These results indicate that the gains mainly come from the cross-domain training, rather than solely the increase of total training samples.

Rotation invariance. We further evaluate the robustness of RAM to large rotations on 3DMatch and 3DLoMatch. Following [4, 20], we apply random rotations sampled from $\mathcal{U}(-\pi, \pi)$. We denote the original setting as standard and the large-rotation setting as rotated. We report RR by directly computing registration RMSE instead of estimating from covariance matrices [7], so the RR values are slightly different those reported in the main paper. For a fair comparison, we evaluate both the standard and rotated settings using the same protocol.

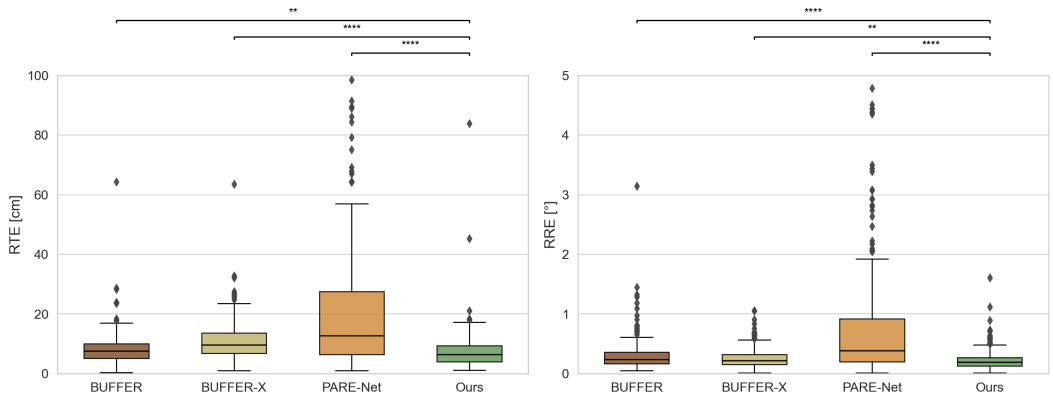
As shown in Tab. 9, PARENet remains stable under large rotations because of its rotation-equivariant design. RAM experiences larger, but still limited, degradation: *Inlier Ratio* decreases by 0.8 and 1.0 points on 3DMatch and 3DLoMatch, respectively, while *Registration Recall* decreases by 0.7 and 2.7 points. Importantly, RAM retains strong absolute performance under large rotations and continues to outperform PARENet in *Inlier Ratio* on both benchmarks. We attribute this robustness to two factors. First, we apply rotation augmentation during training, which improves the efficacy of the KPConv backbone under large rotations. Second, RAM encodes geometric consistency during both superpoint matching and point matching, further mitigating the impact of the rotation-variant backbone.

Cross-source registration. We also evaluate the generalization ability of RAM to cross-source registration on the 3DCSR benchmark [37], following the preprocessing protocols of [38, 39]. As in Tab. 10, although RAM is not trained specifically for cross-source registration, it substantially outperforms GeoTransformer on both subsets, especially on the more challenging LiDAR subset. These results indicate that the cross-domain training strategy also provides meaningful generalization benefits under cross-source shifts.

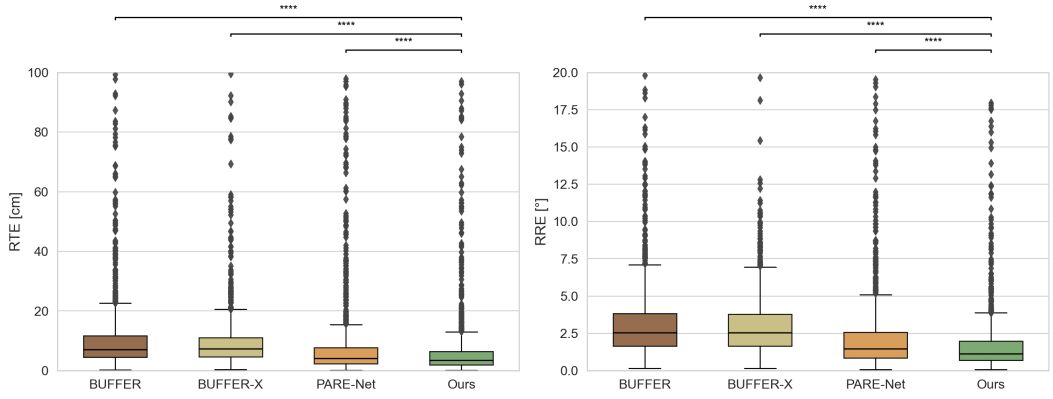
However, RAM remains behind Cross-PCR, which is specifically designed for cross-source registration. We attribute part of this gap to the sensitivity of our current encoder to severe density and sampling discrepancies between heterogeneous sources. Extending RAM with cross-source registration models may further improve its robustness to heterogeneous modalities. However, this is beyond the scope of this work and we will leave it as future work.

Table 10: Results on 3DCSR benchmark. **Boldfaced** numbers highlight the best and the second best are underlined.

Model	SfM		LiDAR	
	IR(%)	RR(%)	IR(%)	RR(%)
GeoTrans [3]	40.2	84.4	3.4	9.1
Cross-PCR [39]	**89.4**	**93.8**	**35.5**	**66.7**
RAM (*ours*)	44.9	90.6	13.3	42.9



(a) ScanNet++



(b) Oxford

Figure 4: Statistical tests on ScanNet++ and Oxford benchmarks.

Statistical test. Following [13], we conduct statistical comparisons on ScanNet++ and Oxford against BUFFER, BUFFER-X, and PARENet. For ScanNet++, we use the official baseline checkpoints trained on 3DMatch. For Oxford, we use the corresponding checkpoints trained on KITTI. As shown in Fig. 4, RAM achieves the lowest median *RTE* and *RRE* with the smallest variance. And the statistical tests confirm that the improvements over all baselines are highly significant.

A.5 Discussions

Trade-off between generalization and per-domain accuracy. To examine the influence of cross-domain training on per-domain accuracy, Tab. 4 (a) of the main paper compares the hybrid model with domain-specific models using the same architecture and optimization protocol. The hybrid model consistently outperforms the corresponding domain-specific models across the evaluated benchmarks. These results indicate that, under our training design, cross-domain training does not introduce a systematic trade-off between generalization and per-domain accuracy.

At the same time, domain-relevant training data remain important for performance within each domain. As shown in Tab. 2, the complete hybrid model achieves the best overall performance, showing that cross-domain diversity is beneficial rather than detrimental. However, removing nuScenes (marked by $^{\diamond}$) leads to a performance decrease, particularly on MIT, while the outdoor-only model (marked by ^{star}) partially recovers this loss, improving *Inlier Ratio* from 62.5 to 72.3 and RR from 97.4 to 98.7. It also matches the hybrid model without nuScenes on WOD.

These results suggest that cross-domain and domain-relevant data play complementary roles: cross-domain data expose the model to more diverse geometric patterns and improve broad generalization, whereas data from the target domain provide important domain-specific structures and sampling characteristics.

Effect of LiDAR sparsity on low-overlap registration. We believe that the improvement on 3DLoMatch primarily arises from the greater geometric diversity introduced by cross-domain training, rather than simply from a larger number of training-sample exposures. As shown in Tab. 8, extending the training of the indoor-only model to approximately match the total sample exposure of the hybrid model does not close the performance gap on 3DLoMatch. The hybrid model still improves *Inlier Ratio* from 50.3 to 53.7 and *Registration Recall* from 79.2 to 81.1.

Although both sparsity and low overlap lead to incomplete observations, they actually impose distinct difficulties on registration: sparsity weakens the local geometric cues within each point cloud, whereas low overlap reduces the shared structure between the pair and increases correspondence ambiguity in non-overlapping regions. Moreover, most outdoor training pairs still have relatively high overlap. Therefore, introducing sparse LiDAR data does not directly expose the model to the same challenge as low-overlap registration.

Instead, the sparse and incomplete observations in LiDAR data enrich the geometric and sampling patterns during training. This diversity may enhances feature discriminativeness and improves the model's robustness to partial observations, therefore benefiting low-overlap registration indirectly.

A.6 Broader Impacts

This work advances robust cross-domain 3D point cloud registration, enabling accurate spatial alignment across diverse sensors without extensive per-domain fine-tuning or re-training. It can act as a LiDAR odometry component in 3D vision systems and benefit broad applications, such as autonomous navigation, augmented and virtual reality, digital twins, construction monitoring, cultural heritage preservation, etc. Another potential impact is that this work unveils the efficacy of cross-domain training in point cloud processing tasks. The geometry-aware normalization enables processing point clouds from diverse domains in a unified model, which could act as a fundamental component towards point cloud foundation models.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have presented our scope and contributions in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: We do not have theoretical results in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the design of our method in Sec. 3, and further introduce the implementation details and the datasets in the appendix. Code and models are publicly available at `https://github.com/qinzheng93/RegisterAnything`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release code and data at `https://github.com/qinzheng93/RegisterAnything`, and the paper and appendix provide detailed methodology and instructions for reproduction.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We present all the training and testing settings in Sec. 4 and the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We present statistical tests in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the experiments compute resources in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Point cloud registration is a low-level 3D vision task, without any societal impacts. We discuss the practical values of our work in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not include data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets are used in accordance with their respective licenses and properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new assets at this time. The code and models will be released at `https://github.com/qinzheng93/RegisterAnything`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is only used for paper editing and formatting purposes, and does not impact the core methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.