R^2: GEOMETRIC REGRESSORS ARE EFFECTIVE POINT CLOUD REGISTRATION LEARNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose R^2, a point cloud registration framework that recasts registration as one-shot shared geometry regression. Instead of learning explicit correspondences or descriptors, R^2 predicts a shared, pair-conditioned canonical space for both partial observations. Using a compact set of geometric anchors, R^2 assigns canonical coordinates to sampled points from both views, including points outside the overlapping regions. Correspondences are then derived naturally from the predicted canonical geometry through mutual nearest-neighbor matching, and the rigid transformation is recovered in the original Euclidean space using a closed-form SVD-based Kabsch solver. To improve canonicalization under partial overlap, we introduce a cross-view geometric coupling loss that complements the coordinate regression loss with relational supervision. The resulting framework requires only pairwise pose supervision to construct canonical targets and does not rely on annotated point-wise correspondence labels. Experiments on 3DMatch and 3DLoMatch demonstrate that R^2 achieves competitive registration accuracy with a simple RANSAC-free inference pipeline and favorable speed–accuracy trade-offs.

1 INTRODUCTION

Point cloud registration is a fundamental problem in 3D perception. It aligns 3D observations acquired from different viewpoints and sampling conditions into a common coordinate frame, and underpins applications such as 3D reconstruction (Monji-Azad et al., 2023), robotic navigation, SLAM, and autonomous driving (Zheng et al., 2022; Shi et al., 2023). Although point clouds provide explicit 3D coordinates, different observations are typically incomplete and non-identical samples of the same scene. Occlusion, sparse sampling, and density variation further complicate cross-view geometric association. As a result, point cloud registration has long been formulated as a correspondence discovery problem: establish cross-view correspondences first, and then recover the rigid transformation under geometric constraints (Besl & McKay, 1992; Du et al., 2007; Zhang et al., 2021; Aoki et al., 2019; Huang et al., 2021; Qin et al., 2023).

Recent advances in image-based 3D reconstruction suggest another perspective. Methods such as DUSt3R (Wang et al., 2024), VGGT (Wang et al., 2025a), and π^3 (Wang et al., 2025b) directly predict 3D geometric representations through feed-forward networks, reducing reliance on explicit correspondence search and back-end optimization. These developments motivate treating shared scene geometry itself as the prediction target for point cloud registration. Here, the geometry of each observation is already available; the remaining task is to infer how the two partial observations fit together. This suggests learning to place both observations into a shared coordinate frame, so that their predicted geometry provides the basis for registration.

Predictive formulations have also been explored within point cloud registration. REGTR (Yew & Lee, 2022) directly regresses cross-view correspondence coordinates, demonstrating that registration can be performed without descriptor matching. More recent methods, including RPF (Sun et al., 2025) and RAP (Pan et al., 2026), generate aligned point configurations through learned geometric flows, but recover the final geometry progressively through multiple integration steps. These approaches motivate a formulation that combines shared geometric prediction with single-pass inference. Rather than predicting correspondence coordinates in the opposite view or progressively transporting points toward alignment, we ask: *Can a feed-forward network jointly regress both partial observations into a shared geometric configuration in a single forward pass?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

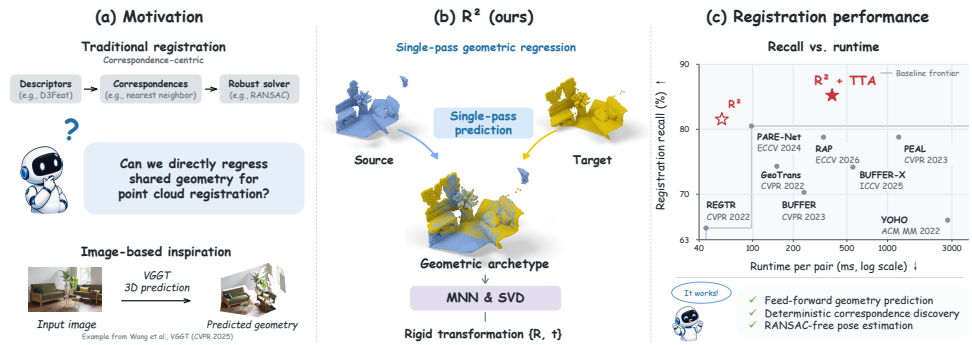

Figure 1: From descriptor matching to shared geometry regression. (a) Motivation: conventional descriptor-based registration and feed-forward 3D prediction from images. (b) R^2 predicts canonical coordinates for both input point clouds in a shared, pair-conditioned frame, then recovers the pose through MNN matching and the Kabsch solver. (c) Registration recall versus runtime on 3DLoMatch.

Despite the availability of input geometry, predicting a shared configuration remains challenging. First, point clouds contain numerous irregularly sampled points. We must efficiently organize them into informative units that preserve local geometry and position for relational reasoning. Second, limited overlap and the absence of dense appearance cues provide little direct evidence for learning how the two observations fit together. We must therefore learn cross-view geometric relations that support a consistent shared configuration.

To address these challenges, we propose R^2, a one-shot shared geometry regression framework. We organize each observation into compact geometric anchors combining sparse local features and positional encoding, enabling efficient within-view and joint attention to predict coordinates in a pair-conditioned canonical space. To learn consistent cross-view structure from limited shared evidence, we combine a coordinate regression loss over all anchors, including non-overlapping regions, with a cross-view geometric coupling loss that preserves local angular relations. At inference, mutual nearest-neighbor (MNN) matching in the predicted space establishes correspondences, and an SVD-based Kabsch solver recovers the pose from matched input points (MNN+SVD), without RANSAC. Explicit predictions enable rigidity-guided test-time augmentation for improved robustness.

Our main contributions are as follows:

- We recast point cloud registration as feed-forward shared geometry regression, demonstrating that shared scene geometry can be directly regressed from partially overlapping point clouds in a single forward pass.
- We introduce a cross-view geometric coupling loss that constrains local angular relations across observations. By supervising the relative arrangement of canonical predictions alongside individual coordinates, it promotes geometry that supports reliable correspondence recovery, particularly under limited overlap.
- We demonstrate competitive single-pass registration performance on 3DMatch and 3DLoMatch, with further gains in registration recall through rigidity-guided test-time augmentation. These results support feed-forward shared geometry regression as an effective approach to registration.

2 Related Work

Descriptor-based matching. Descriptor-based methods establish cross-view correspondences through feature similarities and recover the rigid transformation from the matched points. Early approaches use handcrafted descriptors (Rusu et al., 2009; Tombari et al., 2010) or learned local representations (Zeng et al., 2017; Khoury et al., 2017; Gojcic et al., 2019; Ao et al., 2021), while scene-wise networks support efficient feature extraction and coarse-to-fine matching (Choy et al., 2019; Bai et al., 2020; Huang et al., 2021; Yu et al., 2021; 2023). Subsequent work improves matching through rotation-invariant/equivariant representations (Wang et al., 2022; 2023), cross-view attention (Chen et al., 2024; Fu et al., 2025), multi-domain context integration (Lin et al., 2026), and broader generalization (Ao et al., 2023; Seo et al., 2025). GeoTransformer (Qin et al., 2023) encodes pairwise distances and triplet angles to learn geometric features for matching. Our method instead matches points through predicted coordinates in a shared space.

Direct registration and correspondence regression. Learning-based alternatives to descriptor matching include feature-metric alignment (Aoki et al., 2019; Li et al., 2021), pose estimation from soft correspondences (Wang & Solomon, 2019a;b; Yew & Lee, 2020), and probabilistic registration with learned Gaussian mixtures (Yuan et al., 2020). REGTR (Yew & Lee, 2022) predicts cross-view corresponding positions and overlap scores for RANSAC-free registration, while UniCorrn (Goswami et al., 2026) unifies query-based correspondence prediction across 2D and 3D modalities. FUSER (Jiang et al., 2026) predicts multiview poses in a feed-forward pass, with optional diffusion refinement. R^2 instead jointly predicts both views in a pair-conditioned canonical space.

Geometric prediction and generation. Image-based methods such as DUSt3R (Wang et al., 2024), VGGT (Wang et al., 2025a), and π^3 (Wang et al., 2025b) directly predict 3D geometry through feed-forward networks, motivating shared geometry prediction for point cloud registration. In the point cloud domain, Rectified Point Flow (Sun et al., 2025) learns point-wise velocity fields for registration and assembly, while RAP (Pan et al., 2026) extends flow-based geometric generation to scalable multi-view registration. These methods recover aligned configurations through multiple flow integration steps; related diffusion approaches also use iterative generation for pose estimation (Yuan et al., 2025; Wu et al., 2023; Jiang et al., 2023). For pairwise registration, R^2 predicts the shared configuration in one forward pass and recovers the pose deterministically via MNN+SVD.

3 Preliminaries

3.1 Problem Statement

We consider pairwise rigid point cloud registration. Each point cloud $\mathcal{P}^v = \{\mathbf{p}_i^v\}_{i=1}^{N_v} \subset \mathbb{R}^3$ is associated with a view index $v \in \{s, t\}$, where s and t denote the source and target views. The goal is to estimate a rigid transformation $\mathbf{T}_{s\to t} = (\mathbf{R}, \mathbf{t}) \in SE(3)$ satisfying

$$\min_{\mathbf{R}\in SO(3),\, \mathbf{t}\in\mathbb{R}^3} \sum_{(i,j)\in\mathcal{C}} \left\|\mathbf{R}\mathbf{p}_i^s + \mathbf{t} - \mathbf{p}_j^t\right\|_2^2, \tag{1}$$

where $\mathcal{C}$ denotes the unknown cross-view correspondence set. While descriptor-based methods establish correspondences through feature matching, we recover them from a predicted shared geometric configuration of the two observations.

3.2 From Correspondences to Canonicalization

For each training pair, we jointly center and scale the original point clouds to obtain normalized coordinates $\bar{\mathbf{p}}_i^v$. Using the corresponding ground-truth rotation $\mathbf{R}^*$ and normalized translation $\bar{\mathbf{t}}^*$, we map both observations into a shared canonical frame. We split the relative translation symmetrically to place the shared origin midway between the two view origins, balancing their translational displacements toward the canonical frame. Taking the source as reference, we define

$$\mathbf{x}_i^{s*} = \bar{\mathbf{p}}_i^s + \tfrac{1}{2}(\mathbf{R}^*)^\top \bar{\mathbf{t}}^*, \qquad \mathbf{x}_j^{t*} = (\mathbf{R}^*)^\top \big(\bar{\mathbf{p}}_j^t - \tfrac{1}{2}\bar{\mathbf{t}}^*\big). \tag{2}$$

For a target reference, both canonical targets are additionally rotated by $\mathbf{R}^*$; construction details are provided in Appendix A.1. Although these targets can be constructed directly from ground-truth poses, predicting them from partially overlapping observations remains challenging. Each input already provides its observed 3D geometry; the central difficulty is learning how the two observations fit together when shared evidence is scarce. REGTR (Yew & Lee, 2022) handles partial overlap through overlap-weighted correspondence supervision and confidence-weighted pose estimation. Our insight is that reliable registration depends on both the reliability of individual predictions and their relative arrangement. We therefore combine coordinate supervision over all sampled anchors, $\mathcal{L}_{\text{reg}}$ (Eq. 8), with cross-view geometric coupling, $\mathcal{L}_{\text{coup}}$ (Eq. 10), without introducing an overlap-confidence prediction objective. By explicitly supervising local angular relations across views, coupling makes their relative arrangement a direct learning objective.

4 R^2

We present R^2, a pairwise point cloud registration framework based on one-shot shared geometry regression. Figure 2 summarizes the anchor-based architecture, coordinate regression and geometric coupling objectives, and pose recovery procedure.

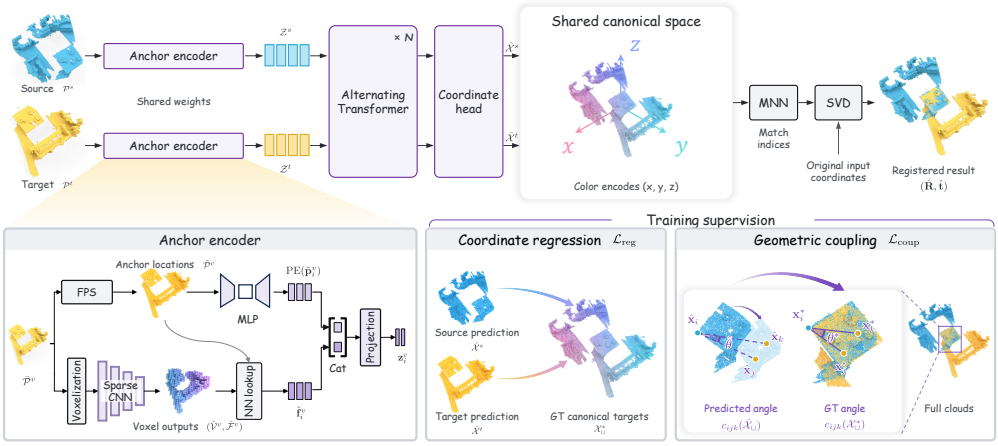

Figure 2: **Overview of R^2.** Shared anchor encoders combine geometric features with positional embeddings at sampled points. An alternating Transformer and coordinate head regress anchor coordinates into a pair-conditioned canonical space. MNN matching establishes correspondences, while SVD recovers the pose from matched input coordinates. The lower panels detail the anchor encoder and training objectives: coordinate regression $\mathcal{L}_{\text{reg}}$ and cross-view geometric coupling $\mathcal{L}_{\text{coup}}$.

4.1 Anchor-based Canonical Regression

Anchor Encoder. Canonical coordinate regression requires a representation different from descriptor learning. While descriptors are optimized to capture discriminative local patterns for matching, canonical regression predicts continuous 3D coordinates and therefore requires each prediction unit to preserve both local geometric context and spatial identity. We define such prediction units as geometric anchors. Each anchor consists of a continuous 3D location and an associated local geometric feature, serving as the basic unit for canonical coordinate regression.

For each normalized point cloud $\bar{\mathcal{P}}^v = \{\bar{\mathbf{p}}_j^v\}_{j=1}^{N_v}$, FPS selects indices a_i^v and anchor locations $\tilde{\mathcal{P}}^v = \{\tilde{\mathbf{p}}_i^v = \bar{\mathbf{p}}_{a_i^v}^v\}_{i=1}^{n_v}$. Below, $\mathbf{x}_i^{v,*}$ denotes the canonical target of the same sampled anchor. FPS provides broad spatial coverage of the input geometry while reducing the number of regression sites compared with dense point-wise prediction. The number of anchors can be adjusted to trade off geometric coverage and computational cost. To provide local geometric context for each anchor, we extract voxel-level features using a sparse 3D encoder. A sparse CNN with voxel size δ produces features and their representative locations:

$$(\hat{\mathcal{V}}^v, \hat{\mathcal{F}}^v) = \mathcal{E}_{\text{sp}}(\bar{\mathcal{P}}^v; \delta), \quad \hat{\mathcal{V}}^v = \{\hat{\mathbf{q}}_m^v\}_{m=1}^{M_v}, \quad \hat{\mathcal{F}}^v = \{\hat{\mathbf{f}}_m^v\}_{m=1}^{M_v}. \tag{3}$$

Here M_v is the number of output voxels, and $\hat{\mathbf{q}}_m^v$ is the input point nearest to voxel m's center, paired with its feature $\hat{\mathbf{f}}_m^v$. Features are assigned to anchors by nearest-representative lookup:

$$\tilde{\mathbf{f}}_i^v = \hat{\mathbf{f}}_{\text{NN}(\tilde{\mathbf{p}}_i^v, \hat{\mathcal{V}}^v)}^v. \tag{4}$$

Each anchor receives a two-layer MLP positional embedding and a learned reference-view embedding:

$$\mathbf{z}_i^v = \text{Proj}([\tilde{\mathbf{f}}_i^v; \text{PE}(\tilde{\mathbf{p}}_i^v)]) + \mathbf{e}_{\text{ref}}(\mathbf{1}[v = r]), \tag{5}$$

Here r is the reference view, Proj is a learned linear projection, and $\mathbf{e}_{\text{ref}}$ encodes whether $v = r$. The operator NN returns a nearest-point index. The resulting anchor tokens $\mathcal{Z}^v = \{\mathbf{z}_i^v\}_{i=1}^{n_v}$ are then used for cross-view interaction and canonical coordinate regression.

Cross-view Canonical Regression. The anchor tokens from both views are jointly processed to predict coordinates in the shared canonical space. The Transformer alternates between within-view attention and joint attention over both views:

$$(\mathcal{H}^s, \mathcal{H}^t) = \text{AltAttn}(\mathcal{Z}^s, \mathcal{Z}^t), \tag{6}$$

Within-view attention aggregates context, while joint attention exchanges information across views. The features $\mathcal{H}^v = \{\mathbf{h}_i^v\}_{i=1}^{n_v}$ are mapped to canonical coordinates by a regression head:

$$\hat{\mathbf{x}}_i^v = \text{Reg}(\mathbf{h}_i^v), \qquad v \in \{s, t\}. \tag{7}$$

The predicted coordinates $\hat{\mathcal{X}}^v$ form the shared canonical representation for correspondence recovery.

4.2 Coordinate Regression and Geometric Coupling

For $n_\cup = n_s + n_t$ anchors, the coordinate regression loss $\mathcal{L}_{\text{reg}}$ averages squared errors over all points and coordinate components:

$$\mathcal{L}_{\text{reg}} = \frac{1}{3n_\cup} \sum_{v\in\{s,t\}} \sum_{i=1}^{n_v} \left\| \hat{\mathbf{x}}_i^v - \mathbf{x}_i^{v,*} \right\|_2^2. \tag{8}$$

This objective directly anchors each prediction to its target canonical position. However, its effectiveness depends on the amount of shared geometry available between the two observations, as observed empirically. While coordinate regression can recover a consistent configuration when sufficient overlap exists, it becomes more challenging under limited overlap, where scarce shared geometric evidence makes it harder to infer how the two views should be arranged in the shared space. The limitation is that $\mathcal{L}_{\text{reg}}$ penalizes coordinate errors for each anchor independently. *It specifies where each anchor should be placed, but does not explicitly penalize errors in geometric relationships between anchors across views.* Such relational structure is important for correspondence recovery, since small coordinate errors can still lead to inconsistent canonical configurations.

We therefore introduce a cross-view geometric coupling loss $\mathcal{L}_{\text{coup}}$ to provide explicit relational supervision beyond point-wise coordinate regression. We construct a KNN graph on the ground-truth canonical targets from both views and retain valid cross-view edges. Let $\mathcal{X}_\cup^*$ and $\hat{\mathcal{X}}_\cup$ denote the consistently indexed concatenations of target and predicted anchors. These heterogeneous edges capture locally shared geometric structures without requiring explicit point-to-point correspondence annotations. We further form triplets from pairs of heterogeneous edges sharing the same reference point. For each triplet (i, j, k), with reference point i and opposite-view neighbors j, k, we characterize their angular relation using a stabilized cosine:

$$c_{ijk}(\mathcal{X}) = \frac{\langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{x}_k - \mathbf{x}_i \rangle}{\|\mathbf{x}_j - \mathbf{x}_i\|_2 \|\mathbf{x}_k - \mathbf{x}_i\|_2 + \epsilon}. \tag{9}$$

where $\epsilon > 0$ is a small constant for numerical stability. The coupling loss encourages the predicted canonical space to preserve these local geometric relations:

$$\mathcal{L}_{\text{coup}} = \mathbb{E}_{(i,j,k)\sim p_{\text{sample}}} \left[\phi_\tau \Big(c_{ijk}(\hat{\mathcal{X}}_\cup) - c_{ijk}(\mathcal{X}_\cup^*) \Big) \right], \tag{10}$$

Here ϕ_τ applies a squared penalty only to the residual magnitude exceeding tolerance τ. The sampling distribution selects valid anchors uniformly and two distinct valid opposite-view neighbors uniformly per anchor. Appendix A.3 defines ϕ_τ and specifies neighborhood validity and empty-set handling. At each supervised layer, the objective combines the coordinate regression and coupling losses:

$$\mathcal{L} = \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} + \lambda_{\text{coup}} \mathcal{L}_{\text{coup}}. \tag{11}$$

To provide intermediate supervision during canonicalization, we introduce Deep Canonical Supervision (DCS). Both objectives are also applied at layers 4, 8, and 12. See Appendix A.3.

4.3 From Regression to Registration

Given the predicted canonical coordinates $\hat{\mathcal{X}}^s = \{\hat{\mathbf{x}}_i^s\}_{i=1}^{n_s}$ and $\hat{\mathcal{X}}^t = \{\hat{\mathbf{x}}_j^t\}_{j=1}^{n_t}$, we recover the source-to-target transform $\hat{\mathbf{T}}_{s\to t} = (\hat{\mathbf{R}}, \hat{\mathbf{t}}) \in SE(3)$.

Canonical-space matching. We first establish candidate correspondences by mutual nearest neighbors (MNN) in the predicted canonical space:

$$\hat{\mathcal{C}} = \Big\{ (i,j) \;\Big|\; j = \text{NN}(\hat{\mathbf{x}}_i^s, \hat{\mathcal{X}}^t),\ i = \text{NN}(\hat{\mathbf{x}}_j^t, \hat{\mathcal{X}}^s) \Big\}. \tag{12}$$

Thus, correspondences are obtained from the predicted canonical coordinates, rather than from descriptor similarities.

Closed-form pose recovery. With normalization center $\mathbf{c}$ and scale α, the original sampled coordinates are $\mathbf{p}_{a_i^v}^v = \alpha \tilde{\mathbf{p}}_i^v + \mathbf{c}$. Using $\hat{\mathcal{C}}$, we recover the pose in the original coordinate units:

$$\hat{\mathbf{T}}_{s\to t} = \arg\min_{\mathbf{T}\in SE(3)} \sum_{(i,j)\in\hat{\mathcal{C}}} \left\| \mathbf{T} \circ \mathbf{p}_{a_i^s}^s - \mathbf{p}_{a_j^t}^t \right\|_2^2, \tag{13}$$

where $\mathbf{T} \circ \mathbf{p} = \mathbf{R}\mathbf{p} + \mathbf{t}$. We solve this objective with the Kabsch solver. No RANSAC-style sampling or iterative pose refinement is used in our default inference pipeline.

Table 1: Indoor results on 3DMatch and 3DLoMatch. Runtime includes model inference and pose estimation, excluding preprocessing. n/a denotes unreported results or unavailable reproducible implementations. † indicates results obtained with additional test-time enhancement strategies.

Group	Method	3DMatch				3DLoMatch			
		RRE (°) ↓	RTE (m) ↓	RR (%) ↑	Time (ms) ↓	RRE (°) ↓	RTE (m) ↓	RR (%) ↑	Time (ms) ↓
RANSAC-based ♣	♣ D3Feat (Bai et al., 2020)	2.161	0.067	81.60	412	3.361	0.103	37.20	398
	♣ FCGF (Choy et al., 2019)	1.949	0.066	85.10	129	3.147	0.100	40.10	131
	♣ Predator (Huang et al., 2021)	2.029	0.064	89.00	626	3.048	0.093	59.80	459
	♣ GeoTrans (Qin et al., 2022)	n/a	n/a	92.00	174	n/a	n/a	75.00	151
	♣ BUFFER (Ao et al., 2023)	n/a	n/a	92.10	245	n/a	n/a	70.30	240
	♣ BUFFERX (Seo et al., 2025)	1.790	0.058	95.58	552	2.839	0.096	74.18	553
RANSAC-free ♢	♢ REGTR (Yew & Lee, 2022)	1.567	0.049	92.00	45	2.827	0.077	64.80	45
	♢ YOHO (Wang et al., 2022)	n/a	n/a	90.80	2932	n/a	n/a	66.00	2788
	♢ PEAL (Yu et al., 2023)	1.748	0.062	94.20	1428	2.825	0.085	78.80	1202
	♢ PARE-Net (Yao et al., 2024)	1.888	0.061	95.00	101	3.084	0.090	80.50	98
	♢ PSReg (Huang et al., 2025)	n/a	n/a	95.70	n/a	n/a	n/a	79.30	n/a
	♢ RAP † (Pan et al., 2026)	n/a	n/a	95.90	372	n/a	n/a	78.78	335
	♢ Ours	1.595	0.056	95.28	59	2.761	0.085	81.63	59
	♢ Ours†	1.601	0.055	**96.41**	169	2.799	0.087	**85.32**	388

5 EXPERIMENTS

We evaluate registration accuracy and efficiency on indoor and outdoor benchmarks in Section 5.2. Section 5.3 examines the contributions of network architecture, training objectives, and pose recovery. Finally, Section 5.4 investigates the registration potential of canonical geometry, the role of geometric coupling, and the practical benefits of canonicalization.

5.1 EXPERIMENTAL SETUP

Datasets and metrics. We evaluate on 3DMatch (Zeng et al., 2017), 3DLoMatch (Huang et al., 2021), and KITTI Odometry (Geiger et al., 2012). 3DMatch contains 62 indoor RGB-D scenes, with 46/8/8 scenes used for training/validation/testing. Its test pairs have more than 30% overlap; 3DLoMatch contains test pairs with 10–30% overlap. We use all 1,279/1,726 official indoor evaluation pairs and report registration recall (RR) at the benchmark RMSE threshold of 0.2 m, relative rotation error (RRE), and relative translation error (RTE). RR is averaged equally over eight scenes. RRE/RTE are first averaged over successful pairs within each scene and then equally over scenes, following the GeoTransformer evaluation code (Qin et al., 2023). For KITTI, sequences 0–5/6–7/8–10 are used for training/validation/testing. We report rotation error (RE), translation error (TE), and transformation recall (TR), using thresholds of 5° and 2 m for TR.

Implementation and inference. Our implementation uses PyTorch (Paszke et al., 2019), spconv (Yan et al., 2018), and FlashAttention (Dao et al., 2022). Experiments use an RTX 3090 GPU and an Intel Xeon Silver 4314 CPU. Following the sampling budget of BUFFER-X (Seo et al., 2025), we use farthest point sampling (FPS) to select 1,500 points per view for indoor training and evaluation, with MNN+SVD for pose recovery unless otherwise stated. Runtime includes network inference and pose recovery, excluding preprocessing; TTA timing also includes repeated passes and rigidity checks. Runtime is measured on the same hardware for all methods with available reproducible implementations. Missing entries reflect unreported results or the lack of an implementation suitable for reproduction. Appendices A and B provide method details and experimental protocols, respectively.

5.2 COMPARISON WITH STATE-OF-THE-ART METHODS

Indoor registration. We first evaluate the effectiveness of shared geometry regression for point cloud registration. As shown in Table 1, R^2 achieves registration recall of 95.28% on 3DMatch and 81.63% on 3DLoMatch using the default MNN+SVD inference pipeline. Compared with REGTR, which directly predicts cross-view correspondence coordinates, R^2 improves recall by 3.28 and 16.83 percentage points on 3DMatch and 3DLoMatch, respectively. The larger improvement on 3DLoMatch suggests that predicting a shared geometric configuration provides more robust registration cues when the overlap between observations is limited. R^2 also achieves a higher recall than PARE-Net on 3DLoMatch (81.63% vs. 80.50%). Compared with iterative geometry generation method RAP, R^2 directly predicts the shared configuration in a single forward pass. It achieves higher recall on 3DLoMatch (81.63% vs. 78.78%) while reducing inference time from 335 ms to 59 ms. These results demonstrate that shared geometry regression can provide effective correspondence cues, enabling accurate registration with a lightweight deterministic pose recovery stage.

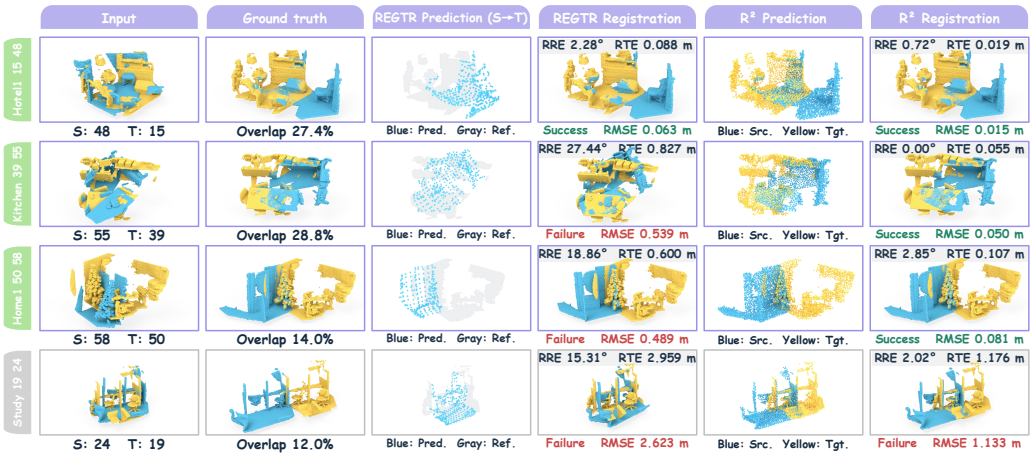

Figure 3: **Prediction spaces and registration outcomes on 3DLoMatch.** REGTR shows all source-to-target predictions; gray points show the target scan, not point-wise ground truth. Its solver weights bidirectional correspondences by predicted overlap. R^2 shows both canonical outputs.

Improving robustness with test-time augmentation. Explicit canonical predictions enable rigidity-guided test-time selection. We apply common rotations to both input views and select predictions by their input-to-prediction rigid fitting residuals. Implementation details are provided in Appendix A.4. As shown in Table 1, this inference-time enhancement improves the registration recall of R^2 from 95.28% to 96.41% on 3DMatch and from 81.63% to 85.32% on 3DLoMatch. The gain is more pronounced on 3DLoMatch, where limited overlap makes canonical prediction more challenging. Since correspondence extraction and pose recovery remain unchanged, the improvement comes from selecting canonical predictions with better geometric consistency rather than modifying the registration pipeline. The improved robustness comes at the cost of increased inference time, resulting in 169 and 388 ms per pair on 3DMatch and 3DLoMatch, respectively.

Qualitative analysis of canonical predictions. We compare R^2 with REGTR, another regression-based method that avoids descriptor matching and RANSAC. REGTR predicts coordinates in the opposite view's frame, whereas R^2 maps both views into a shared canonical space. The two predicted point sets exhibit compatible spatial arrangements, allowing their overlapping regions to support mutual nearest-neighbor matching and accurate pose recovery. These examples illustrate that a shared canonical representation can retain the geometry of both views while making their relative alignment explicit. The final example shows that visually coherent canonical predictions do not necessarily yield a correct relative pose.

Outdoor registration. Table 2 evaluates R^2 on KITTI Odometry to assess canonical regression on outdoor LiDAR observations. R^2 achieves 99.8% transformation recall, with a rotation error of $0.24°$ and a translation error of 7.7 cm. These results show that the proposed canonical regression framework with MNN+SVD can be applied beyond indoor scenes.

Table 2: Outdoor results on KITTI.

Method	RE (°)	TE (cm)	TR (%)
Predator (Huang et al., 2021)	0.27	6.8	**99.8**
SpinNet (Ao et al., 2021)	0.47	9.9	99.1
CoFiNet (Yu et al., 2021)	0.41	8.2	**99.8**
GeoTrans (Qin et al., 2023)	**0.23**	6.2	**99.8**
BUFFER (Ao et al., 2023)	0.26	7.1	**99.8**
PARE-Net (Yao et al., 2024)	**0.23**	**4.9**	**99.8**
BUFFERX (Seo et al., 2025)	0.27	7.7	**99.8**
Ours	0.24	7.7	**99.8**

5.3 Ablation Studies

We examine how the input representation, training objectives, and pose recovery contribute to registration through shared geometry regression in Table 3.

Network architecture. We first examine two architectural choices: absolute positional encoding and the local feature backbone. Removing absolute positional encoding reduces RR by 1.61/3.89 pp on 3DMatch/3DLoMatch. This suggests that local voxel features alone do not fully preserve the point-level spatial information needed for coordinate regression. Replacing the sparse convolutional backbone with KPConv yields comparable accuracy, suggesting that the regression framework is not strongly tied to either backbone. However, KPConv increases inference time from 59 ms to 108 ms per pair. We therefore adopt sparse convolutions for a better accuracy–efficiency trade-off.

Table 3: **Component ablations of R^2.** Full model: sparse convolutional backbone, absolute positional encoding, $\mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{coup}}$, and MNN+SVD. All entries use FPS1500 without TTA.

Group	Setting	3DMatch				3DLoMatch			
		RRE (°) ↓	RTE (m) ↓	RR (%) ↑	Time (ms) ↓	RRE (°) ↓	RTE (m) ↓	RR (%) ↑	Time (ms) ↓
Network	w/o absolute PE	1.939	0.066	93.67	59	3.499	0.099	77.74	59
	w/ KPConv	1.630	0.057	94.86	108	2.904	0.089	81.55	107
Supervision	w/o $\mathcal{L}_{\text{coup}}$	2.696	0.082	86.02	59	4.176	0.115	52.09	59
	Overlap-weighted $\mathcal{L}_{\text{reg}}$ only	2.107	0.071	93.29	61	3.730	0.103	73.91	61
	w/o DCS	1.801	0.060	94.11	59	2.982	0.091	76.50	59
Pose	Direct rigid fits	1.642	0.061	95.34	60	2.690	0.096	78.44	60
	MNN + RANSAC	1.589	0.055	95.39	62	2.698	0.083	82.17	64
Reference	**Full model**	1.595	0.055	95.28	59	2.761	0.085	81.63	59

Training objectives. We next examine how the choice of supervision objectives and their application at intermediate layers affect registration performance. Using only the coordinate regression loss $\mathcal{L}_{\text{reg}}$ yields limited registration accuracy, particularly on 3DLoMatch. Weighting the coordinate loss by overlap confidence improves performance, suggesting that emphasizing points with shared observations helps the model learn cross-view alignment. However, the full objective achieves better registration by retaining equal-weight coordinate supervision over all sampled points and introducing $\mathcal{L}_{\text{coup}}$ to constrain local angular relations across views. This comparison suggests that, rather than relying solely on overlap weighting to emphasize shared regions, supervising all observed geometry while explicitly constraining the relative arrangement of predictions provides more effective guidance for registration. We further examine DCS by retaining both objectives only at the final Transformer layer. Removing intermediate supervision reduces RR to 94.11%/76.50% on 3DMatch/3DLoMatch, a drop of 1.17/5.13 percentage points relative to the full model (Table 3).

Pose recovery. Finally, we examine how canonical predictions support pose recovery. Directly fitting each input view to its predicted canonical coordinates already achieves strong registration performance, indicating that the predicted geometry itself provides informative pose cues. MNN+SVD further improves performance on 3DLoMatch, highlighting the benefit of cross-view matching under limited overlap. We also replace the Kabsch solver with RANSAC on the same MNN correspondences to test whether pose recovery relies on robust estimation. RANSAC provides only a modest improvement (0.11/0.54 pp on 3DMatch/3DLoMatch). This suggests that strong registration performance does not rely heavily on RANSAC. We therefore retain the deterministic closed-form Kabsch solver by default.

5.4 Analysis of Canonical Regression

To understand why geometric regressors are effective point cloud registration learners, we ask: *(i) How well can canonical geometry support registration? (ii) What makes predicted geometry useful for matching? (iii) What practical benefits does geometric regression offer?*

Canonical-space Oracle. We answer the first question by replacing predicted coordinates with GT canonical coordinates while keeping the sampled inputs and MNN+SVD pipeline fixed. The resulting near-perfect registration on both benchmarks (Figure 4(a)) demonstrates the potential of shared geometry for registration with a simple matching and pose recovery pipeline. Unlike REGTR, which explicitly predicts overlap, R^2 relies on MNN matching in the regressed canonical space to select correspondences. Figure 4(b) shows that MNN matching preferentially retains points in overlapping regions, without an explicit overlap prediction module.

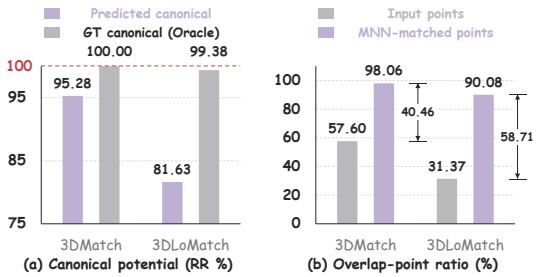

Figure 4: **Canonical-space registration and overlap.** (a) Recall using predicted versus GT canonical coordinates. (b) Overlap-point fractions in inputs and MNN matches, using GT labels for evaluation.

Geometric Coupling Analysis. We answer the second question by comparing training with $\mathcal{L}_{\text{reg}}$ alone against training with both losses. Figure 5(a) shows that geometric coupling helps avoid incorrect cross-view associations caused by locally similar structures, improving canonical alignment

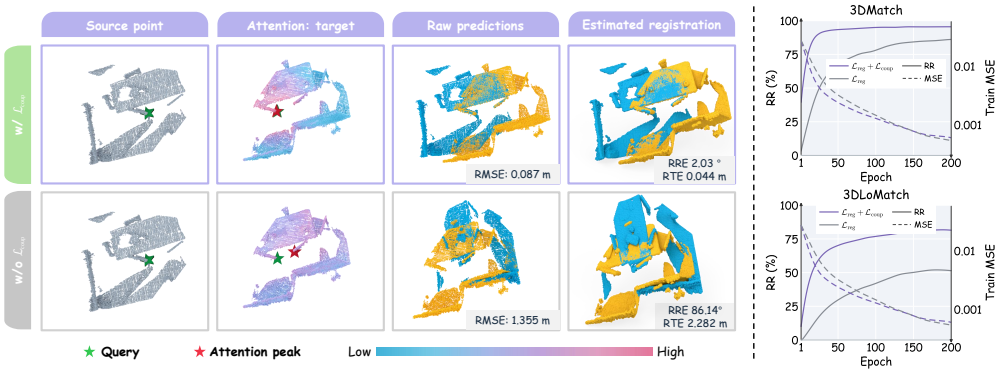

Figure 5: **Geometric coupling shapes geometry for registration.** (a) Top/bottom rows use/omit $\mathcal{L}_{\text{coup}}$. Green/red markers denote the query/attention peak; attention colors encode response strength. Without coupling, attention is diffuse and peaks at a locally similar but incorrect region. Coupling concentrates attention around the correct region and improves alignment. (b) Coordinate-only supervision yields lower training coordinate MSE but worse registration recall.

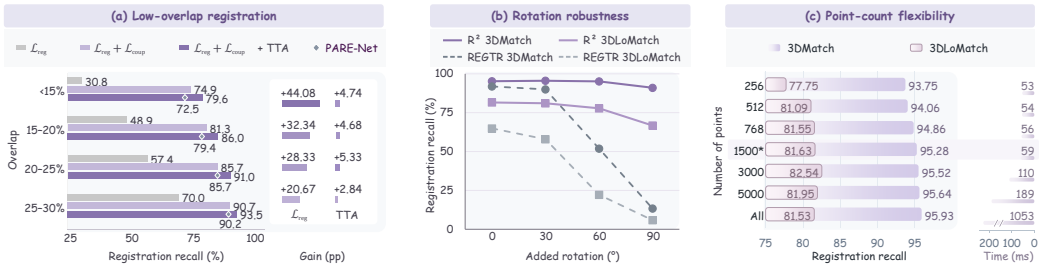

Figure 6: **Canonicalization benefits.** (a) Low-overlap registration. (b) Recall under added source rotations, without TTA. (c) Inference-time point-count flexibility; $*$ marks the training point count.

and pose recovery. The training curves in Figure 5(b) further show that optimizing $\mathcal{L}_{\text{reg}}$ alone yields lower training coordinate MSE but substantially worse registration recall. Thus, average point-wise error does not fully capture the quality of geometry required for matching, which depends on the relative arrangement of predictions across views. Together, these observations suggest that effective geometric regression benefits from supervising both cross-view relations and individual coordinates.

Canonicalization Benefits. We answer the third question through low-overlap registration, robustness to added rotations, and inference flexibility. As overlap decreases, the full model's advantage over coordinate-only supervision grows (Figure 6(a)), supporting cross-view geometric constraints under limited shared evidence. R^2 also matches or exceeds PARE-Net (Yao et al., 2024) across all four low-overlap bins. Its explicit input-to-prediction mapping enables rigidity-guided selection of TTA predictions, further improving low-overlap registration. Under identical added source rotations from $0°$ to $90°$, with updated ground-truth poses and no TTA, our trained model maintains higher recall with less degradation than the released REGTR checkpoint (Figure 6(b)). Interestingly, the model trained with 1,500 anchors per view performs well across inference-time anchor counts (Figure 6(c)). Each anchor retrieves local geometric features from the sparse backbone, allowing anchor counts to vary without retraining. Compact anchor sets already support effective registration; additional anchors bring limited gains at higher computational cost.

6 Conclusion

We presented R^2, a point cloud registration framework that casts pairwise registration as one-shot canonical regression into a pair-conditioned canonical space. With coordinate regression and cross-view geometric coupling losses derived from pairwise poses, R^2 learns view-consistent structure without annotated point-wise correspondence labels. At inference time, MNN correspondences in the canonical space are used with a closed-form Kabsch solver, avoiding RANSAC-style sampling in the default pipeline. Experiments on standard benchmarks show favorable speed–accuracy trade-offs.

AI USE STATEMENT

Generative AI tools were used to correct grammatical errors, improve the wording and readability of the manuscript, and assist with literature searches for recent methods. All AI-assisted content was reviewed by the authors, who take full responsibility for the manuscript, including its claims and references.

ETHICS STATEMENT

This work studies pairwise point cloud registration using existing public indoor and outdoor benchmarks. It does not involve new data collection or experiments with human participants. Applications involving captured environments should respect privacy and the permitted uses of the underlying data.

REPRODUCIBILITY STATEMENT

Section 4 describes the network architecture, training objectives, and inference procedure, while Section 5.1 specifies the benchmarks and evaluation settings. Appendices A and B provide additional method details and experimental protocols, including canonical target construction, preprocessing, training, and evaluation metrics. Source code and pretrained models will be publicly released upon acceptance.

REFERENCES

Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. SpinNet: Learning a general surface descriptor for 3D point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11753–11762, 2021. 2, 7

Sheng Ao, Qingyong Hu, Hanyun Wang, Kai Xu, and Yulan Guo. Buffer: Balancing accuracy, efficiency, and generalizability in point cloud registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1255–1264, 2023. 2, 6, 7

Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & efficient point cloud registration using PointNet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7163–7172, 2019. 1, 3

Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3Feat: Joint learning of dense detection and description of 3D local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6359–6367, 2020. 2, 6, 17

Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pp. 586–606. Spie, 1992. 1

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*, pp. 21698–21707, 2024. 2

Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8958–8966, 2019. 2, 6, 17

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022. 6

Shaoyi Du, Nanning Zheng, Shihui Ying, and Jishang Wei. Icp with bounded scale for registration of md point sets. In *2007 IEEE International Conference on Multimedia and Expo*, pp. 1291–1294. IEEE, 2007. 1

Kexue Fu, Mingzhi Yuan, Changwei Wang, Weiguang Pang, Jing Chi, Manning Wang, and Longxiang Gao. Dual focus-attention transformer for robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11769–11778, 2025. 2

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012. 6

Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3D point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5545–5554, 2019. 2, 17

Prajnan Goswami, Tianye Ding, Feng Liu, and Huaizu Jiang. UniCorrn: Unified correspondence transformer across 2D and 3D. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9943–9954, 2026. 3

Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 4267–4276, 2021. 1, 2, 6, 7, 17

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3788–3796, 2025. 6

Haobo Jiang, Mathieu Salzmann, Zheng Dang, Jin Xie, and Jian Yang. Se 3 diffusion model-based point cloud registration for robust 6d object pose estimation. *Advances in Neural Information Processing Systems*, 36: 21285–21297, 2023. 3

Haobo Jiang, Jin Xie, Jian Yang, Liang Yu, and Jianmin Zheng. FUSER: Feed-forward multiview 3d registration transformer and $\mathrm{SE}(3)^N$ diffusion refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7393–7403, 2026. 3

Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 153–161, 2017. 2

Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Pointnetlk revisited. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12763–12772, 2021. 3

Shuyuan Lin, Wenwu Peng, Junjie Huang, Qiang Qi, Miaohui Wang, and Jian Weng. MCI-Net: A robust multi-domain context integration network for point cloud registration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23585–23593, 2026. 2

Sara Monji-Azad, Juergen Hesser, and Nikolas Loew. A review of non-rigid transformations and learning-based 3d point cloud registration methods. *ISPRS journal of photogrammetry and remote sensing*, 196:58–72, 2023. 1

Yue Pan, Tao Sun, Liyuan Zhu, Lucas Nunes, Iro Armeni, Jens Behley, and Cyrill Stachniss. Register any point: Scaling 3d point cloud registration by flow matching. In *European Conference on Computer Vision (ECCV)*, pp. 21–43, 2026. 1, 3, 6

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11143–11152, 2022. 6

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. 1, 2, 6, 7, 16, 17

Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pp. 3212–3217. IEEE, 2009. 2

Minkyun Seo, Hyungtae Lim, Kanghee Lee, Luca Carlone, and Jaesik Park. Buffer-x: Towards zero-shot point cloud registration in diverse scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3851–3862, 2025. 2, 6, 7, 16

Chenghao Shi, Xieyuanli Chen, Huimin Lu, Wenbang Deng, Junhao Xiao, and Bin Dai. Rdmnet: Reliable dense matching based point cloud registration for autonomous driving. *IEEE Transactions on intelligent transportation systems*, 24(10):11372–11383, 2023. 1

Tao Sun, Liyuan Zhu, Shengyu Huang, Shuran Song, and Iro Armeni. Rectified point flow: Generic point cloud pose estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 1, 3

Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique shape context for 3d data description. In *Proceedings of the ACM workshop on 3D object retrieval*, pp. 57–62, 2010. 2

Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1630–1641, 2022. 2, 6

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 Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10376–10393, 2023. 2

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a. 1, 3

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024. 1, 3

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025b. 1, 3

Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3523–3532, 2019a. 3

Yue Wang and Justin M Solomon. PRNet: Self-supervised learning for partial-to-partial registration. In *Advances in Neural Information Processing Systems*, pp. 8812–8824, 2019b. 3

Yue Wu, Yongzhe Yuan, Xiaolong Fan, Xiaoshui Huang, Maoguo Gong, and Qiguang Miao. Pcrdiffusion: Diffusion probabilistic models for point cloud registration. *arXiv preprint arXiv:2312.06063*, 2023. 3

Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 6

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*, pp. 287–303. Springer, 2024. 6, 7, 9, 17

Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11824–11833, 2020. 3

Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6677–6686, 2022. 1, 3, 6

Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. CoFiNet: Reliable coarse-to-fine correspondences for robust point cloud registration. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 23872–23884, 2021. 2, 7, 17

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17702–17711, 2023. 2, 6, 17

Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *European conference on computer vision*, pp. 733–750. Springer, 2020. 3

Yongzhe Yuan, Yue Wu, Xiaolong Fan, Maoguo Gong, Qiguang Miao, and Wenping Ma. Where precision meets efficiency: Transformation diffusion model for point cloud registration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9734–9742, 2025. 3

Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1802–1811, 2017. 2, 6

Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3450–3466, 2021. 1

Yuchao Zheng, Yujie Li, Shuo Yang, and Huimin Lu. Global-pbnet: A novel point cloud registration for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):22312–22319, 2022. 1

This appendix provides method details, experimental protocols, supplementary results, an analysis of correspondence stability, and limitations.

CONTENTS

A. Method Details 14
A.1 Canonical Target Construction 14
A.2 Anchor Representation and Network Architecture 14
A.3 Geometric Coupling and Deep Canonical Supervision 14
A.4 Pose Recovery and Rigidity-guided TTA 15
B. Experimental Protocols 15
B.1 Datasets and Preprocessing 15
B.2 Training Details 15
B.3 Evaluation Metrics and Runtime Measurement 16
C. Additional Results and Analyses 17
C.1 Scene-wise Registration Results 17
C.2 Registration across Overlap Levels 17
C.3 Canonical Geometry and Registration Quality 17
C.4 Qualitative Results and Failure Cases 19
D. Correspondence Stability and Pose Recovery 20
D.1 Correspondence Stability under Bounded Prediction Errors 20
D.2 Pose Recovery from Correct Correspondences 21
E. Limitations and Reproducibility 21
E.1 Limitations and Future Work 21
E.2 Code and Model Availability 21

A METHOD DETAILS

A.1 CANONICAL TARGET CONSTRUCTION

Table 4 details the construction from preprocessed input clouds and their GT relative pose, using column vectors and $\mathbf{p}^t = \mathbf{R}^*\mathbf{p}^s + \mathbf{t}^*$ for ideal correspondences. The view with more loaded points defines the reference r (source on ties); preprocessing details are given in Appendix B.1.

Table 4: **Canonical target construction** for source and target reference views.

Step	Definition
Shared center	$\boldsymbol{\mu}_v = \frac{1}{N_v}\sum_{i=1}^{N_v}\mathbf{p}_i^v$; $\mathbf{c} = \frac{1}{2}(\boldsymbol{\mu}_s + \boldsymbol{\mu}_t)$.
Shared scale	$\alpha = \max_{d\in\{1,2,3\}}\left(\max_i[\mathbf{p}_i^r]_d - \min_i[\mathbf{p}_i^r]_d\right) + 10^{-6}$.
Normalized coordinates	$\bar{\mathbf{p}}_i^v = (\mathbf{p}_i^v - \mathbf{c})/\alpha$, $v \in \{s, t\}$.
Normalized GT pose	Rotation: $\mathbf{R}^*$; translation: $\bar{\mathbf{t}}^* = [\mathbf{t}^* + (\mathbf{R}^* - \mathbf{I})\mathbf{c}]/\alpha$.
Targets if $r = s$	$\mathbf{x}_i^{s*} = \bar{\mathbf{p}}_i^s + \frac{1}{2}(\mathbf{R}^*)^\top\bar{\mathbf{t}}^*$; $\mathbf{x}_j^{t*} = (\mathbf{R}^*)^\top(\bar{\mathbf{p}}_j^t - \frac{1}{2}\bar{\mathbf{t}}^*)$.
Targets if $r = t$	$\mathbf{x}_i^{s*} = \mathbf{R}^*\bar{\mathbf{p}}_i^s + \frac{1}{2}\bar{\mathbf{t}}^*$; $\mathbf{x}_j^{t*} = \bar{\mathbf{p}}_j^t - \frac{1}{2}\bar{\mathbf{t}}^*$.

The shared normalization updates the translation so that $\bar{\mathbf{p}}^t = \mathbf{R}^*\bar{\mathbf{p}}^s + \bar{\mathbf{t}}^*$ remains valid. The reference view fixes the canonical orientation, while the translation is split symmetrically between the two views. Consequently, ideal corresponding points receive identical canonical targets, and every sampled anchor inherits its input point's target, including anchors outside the overlap. The GT pose is required only for target construction during training.

A.2 ANCHOR REPRESENTATION AND NETWORK ARCHITECTURE

Each geometric anchor combines a sampled input location with a local feature retrieved from the sparse backbone, as described in Section 4.1. Given the input point coordinates, we first voxelize the point cloud to construct sparse coordinates, while a two-layer MLP is used to lift each 3D point coordinate into an initial feature space. The resulting sparse tensor is then processed by two sparse residual blocks at the original resolution, followed by three downsampling stages. Each stage consists of a $3 \times 3 \times 3$ SparseConv3d layer with stride 2 and padding 1, followed by batch normalization, ReLU, and two sparse residual blocks. This design progressively enlarges the receptive field and aggregates local geometric context through overlapping sparse convolutions.

To maintain the correspondence between sparse voxels and the original point set, we associate each output voxel with the input point nearest to its voxel center. This representative-point selection allows the encoder to output not only downsampled sparse features, but also the corresponding downsampled point coordinates and point indices with respect to the original input, which facilitates subsequent point-level geometric processing. In our implementation, the channel dimensions are set to $(32, 64, 128, 256)$, the downsampling strides are $(2, 2, 2)$, and feature assignment to each sampled anchor uses nearest-neighbor lookup rather than local KNN aggregation.

The sampled anchor location is encoded by a two-layer positional MLP and combined with its retrieved feature. The resulting tokens interact through 16 Transformer blocks alternating between within-view and joint attention, with feature dimension 512, eight attention heads, and feed-forward dimension 512. The same coordinate regression head structure is applied across anchors, so their number can vary at inference time.

A.3 GEOMETRIC COUPLING AND DEEP CANONICAL SUPERVISION

For the indoor regression-plus-coupling configuration, we use $\lambda_{\text{reg}} = 10$ and $\lambda_{\text{coup}} = 1$. The coordinate regression loss $\mathcal{L}_{\text{reg}}$ is averaged over points and coordinate components, as in Eq. 8. We construct a $K = 64$ nearest-neighbor graph on GT canonical anchor coordinates and retain local cross-view edges. We use $\epsilon = 10^{-12}$ in the cosine denominator and a cosine-residual tolerance

$\tau = 10^{-3}$. The tolerance function in Eq. 10 is $\phi_\tau(u) = \max(|u| - \tau, 0)^2$. Thus cosine differences within the tolerance incur no penalty, while the excess is penalized quadratically.

An anchor is valid only if it has at least two valid neighbors from the opposite view. For each valid anchor, we draw independent uniform random scores for its valid neighbors and select the two highest-scoring entries. This samples a distinct neighbor pair uniformly without replacement. We average one sampled residual per valid anchor, giving equal weight to valid anchors rather than to all possible triplets. Both source- and target-view anchors participate, and neighborhoods never cross input-pair boundaries. If there are no valid anchors, the coupling loss $\mathcal{L}_{\text{coup}}$ is set to zero while coordinate regression remains active.

DCS. The coordinate regression and coupling losses are also applied to intermediate predictions at Transformer layers 4, 8, and 12, with auxiliary weights 0.025, 0.05, and 0.075, respectively. All supervised layers use the same canonical targets and objective definitions; the final prediction receives the main loss.

A.4 Pose Recovery and Rigidity-guided TTA

Default pose recovery. We establish mutual nearest-neighbor correspondences between the predicted canonical coordinates, then estimate the rigid transformation from the corresponding original input points using a closed-form Kabsch solver. The default pipeline uses one network forward pass and no iterative pose refinement.

Rigidity-guided TTA. We apply common rotations to both input views using per-axis angle magnitudes of $15°$, $30°$, $45°$, and $60°$, with eight sign combinations per magnitude. Each candidate is generated from the original input, giving at most 33 forward passes including the initial prediction. The selection score is the maximum of the two input-to-prediction rigid-fitting RMSEs, without scale fitting. TTA is triggered when the initial score exceeds 0.02 and stops at the first candidate with score at most 0.015; if none qualifies, the candidate with the smallest score is selected. The mean numbers of forward passes are 2.78 and 6.33 on 3DMatch and 3DLoMatch, respectively.

B Experimental Protocols

B.1 Datasets and Preprocessing

We use 3DMatch and 3DLoMatch for indoor registration and KITTI Odometry for outdoor registration, with the splits described in Section 5.1. The indoor evaluation comprises all 1,279 3DMatch pairs and 1,726 3DLoMatch pairs across eight test scenes. KITTI sequences 0–5, 6–7, and 8–10 are used for training, validation, and testing, respectively.

Reference selection uses point counts at loading, before point limiting or augmentation; if the views are swapped, the reference remains attached to the same observation. The clouds and GT pose in Appendix A.1 are taken after any augmentation and before normalization, with N_v denoting the resulting point count.

The two input views undergo the common centering and scale normalization described in Appendix A.1. Sparse voxel features provide local context, while FPS selects the anchor locations used for coordinate prediction. Indoor training and the default reported evaluation use 1,500 anchors per view; the prediction-count analysis varies this budget explicitly. Canonical training targets use the ground-truth relative pose, whereas inference predicts the shared configuration from the observations.

B.2 Training Details

We implement our method in PyTorch and train it on NVIDIA RTX 3090 GPUs. The detailed configurations are summarized in Table 5. The indoor model is trained and evaluated with FPS1500 for the reported results.

Table 5: Detailed implementation settings of our method on 3DMatch and KITTI.

Item	3DMatch	KITTI
Training		
Optimizer	AdamW	AdamW
Batch size	16	16
Initial learning rate	2×10^{-4}	2×10^{-4}
Minimum learning rate	1×10^{-6}	1×10^{-6}
Weight decay	1×10^{-2}	1×10^{-2}
Epochs	200	400
Gradient clipping	1.0	1.0
Geometry / Model		
d_{model}	512	512
Number of heads	8	8
Feed-forward dimension	512	512
Transformer layers	16	16
Voxel size	0.00625	0.0025
Coupling KNN size	64	64
Coupling radius	0.25	0.05

B.3 Evaluation Metrics and Runtime Measurement

Following prior registration works (Seo et al., 2025), we evaluate our method using four standard metrics, including Registration Recall (RR), Relative Rotation Error (RRE), Relative Translation Error (RTE), and Transformation Recall (TR).

Registration Recall (RR). For indoor registration, we follow the GeoTransformer evaluation protocol (Qin et al., 2023). A pair is successfully registered if its benchmark RMSE is below 0.2 m, computed from the relative-transform error and the provided ground-truth information matrix. RR is first computed within each test scene and then averaged equally over the eight scenes:

$$\mathrm{RR} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{M_s}\sum_{i=1}^{M_s}\mathbb{1}[e_{s,i} < 0.2^2], \qquad S = 8, \tag{14}$$

where $e_{s,i}$ is the benchmark evaluator's squared error for pair i in scene s, and M_s is the number of evaluated pairs in that scene.

Relative Rotation Error (RRE). RRE evaluates the angular difference between the estimated rotation and the ground-truth rotation:

$$\mathrm{RRE} = \frac{180}{\pi}\arccos\left(\frac{\mathrm{tr}(\mathbf{R}_{est}^{\top}\mathbf{R}_{gt}) - 1}{2}\right), \tag{15}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator.

Relative Translation Error (RTE). RTE measures the Euclidean distance between the estimated translation vector and the ground-truth translation vector:

$$\mathrm{RTE} = \|\mathbf{t}_{est} - \mathbf{t}_{gt}\|_2. \tag{16}$$

For indoor RRE and RTE, we first average over successfully registered pairs within each scene and then average equally over scenes.

Transformation Recall (TR). TR is defined as the percentage of point cloud pairs whose estimated transformations satisfy both the rotation and translation error thresholds:

$$\mathrm{TR} = \frac{1}{M}\sum_{i=1}^{M}(\mathrm{RRE}_i < \tau_r \text{ and } \mathrm{RTE}_i < \tau_t). \tag{17}$$

For KITTI, we use $\tau_r = 5^\circ$ and $\tau_t = 2\,\mathrm{m}$.

Runtime measurement. Runtime includes network inference and pose recovery, excluding preprocessing and disk I/O. TTA timing also includes the repeated network passes and rigidity checks. Experiments use an RTX 3090 GPU and an Intel Xeon Silver 4314 CPU. The separate timing run uses FPS1500 and averages 2.74 and 6.30 TTA forward passes on 3DMatch and 3DLoMatch, respectively.

C Additional Results and Analyses

C.1 Scene-wise Registration Results

Table 6 reports results with and without TTA using 1,500 anchors per view. TTA improves scene-averaged RR from 95.28% to 96.41% on 3DMatch and from 81.63% to 85.32% on 3DLoMatch. All means weight the eight scenes equally. Rotation and translation errors use each method's own successful pairs, so increased recall need not reduce the average pose error.

Table 6: Scene-wise results on 3DMatch and 3DLoMatch. GeoTransformer results follow PARE-Net (Yao et al., 2024), whereas Table 1 reports the original RANSAC-based results. † denotes TTA.

	3DMatch								3DLoMatch							
Model	Kitchen	Home1	Home2	Hotel1	Hotel2	Hotel3	Study	Lab	Mean	Kitchen	Home1	Home2	Hotel1	Hotel2	Hotel3	Study	Lab	Mean
Registration Recall(%↑)																		
PerfectMatch (Gojcic et al., 2019)	90.6	90.6	65.4	89.6	82.1	80.8	68.4	60.0	78.4	51.4	25.9	44.1	41.1	30.7	36.6	14.0	20.3	33.0
FCGF (Choy et al., 2019)	98.0	94.3	68.6	96.7	91.0	84.6	76.1	71.1	85.1	60.8	42.2	53.6	53.1	38.0	26.8	16.1	30.4	40.1
D3Feat (Bai et al., 2020)	96.0	86.8	67.3	90.7	88.5	80.8	78.2	64.4	81.6	49.7	37.2	47.3	47.8	36.5	31.7	15.7	31.9	37.2
Predator (Huang et al., 2021)	97.6	97.2	74.8	98.9	96.2	88.5	85.9	73.3	89.0	71.5	58.2	60.8	77.5	64.2	61.0	45.8	39.1	59.8
CoFiNet (Yu et al., 2021)	96.4	99.1	73.6	95.6	91.0	84.6	89.7	84.4	89.3	76.7	66.7	64.0	81.3	65.0	63.4	53.4	69.6	67.5
GeoTrans (Qin et al., 2023)	98.4	97.2	83.0	97.8	92.3	88.5	90.6	91.1	92.4	86.5	72.8	70.7	89.0	69.6	73.8	56.1	75.7	74.3
PEAL (Yu et al., 2023)	99.6	98.1	83.6	99.5	96.2	96.2	91.5	88.9	94.2	90.3	75.1	76.1	94.0	79.7	75.5	63.7	75.7	78.8
PARENet (Yao et al., 2024)	99.6	97.2	83.0	99.5	96.2	96.2	91.0	97.8	95.0	91.8	73.1	73.4	94.8	79.0	78.6	66.2	87.1	80.5
R^2	99.1	98.1	80.5	98.4	98.7	100.0	91.9	95.6	95.3	88.4	83.7	76.6	91.9	80.4	83.3	65.8	82.9	81.6
R^2 †	99.8	100.0	83.0	98.9	100.0	100.0	94.0	95.6	96.4	94.1	88.0	79.3	94.8	85.5	83.3	70.5	87.1	85.3
Rotation Error(° ↓)																		
PerfectMatch (Gojcic et al., 2019)	1.926	1.843	2.324	2.041	1.952	2.908	2.296	2.301	2.199	3.020	3.898	3.427	3.196	3.217	3.328	4.325	3.814	3.528
FCGF (Choy et al., 2019)	1.767	1.849	2.210	1.867	1.667	2.417	2.024	1.792	1.949	2.904	3.229	3.277	2.768	2.801	2.822	3.372	4.006	3.147
D3Feat (Bai et al., 2020)	2.016	2.029	2.425	1.990	1.967	2.400	2.346	2.115	2.161	3.226	3.492	3.373	3.330	3.165	2.972	3.708	3.619	3.361
Predator (Huang et al., 2021)	1.861	1.806	2.473	2.045	1.600	2.458	2.067	1.926	2.029	3.079	2.637	3.220	2.694	2.907	3.390	3.046	3.412	3.048
CoFiNet (Yu et al., 2021)	1.910	1.835	2.316	1.767	1.753	1.639	2.527	2.345	2.011	3.213	3.119	3.711	2.842	2.897	3.194	4.126	3.138	3.280
GeoTrans (Qin et al., 2023)	2.460	1.523	2.043	1.572	1.528	1.691	2.149	1.978	1.868	3.246	2.646	3.156	2.769	2.741	2.719	3.338	3.387	3.000
PEAL (Yu et al., 2023)	1.798	1.488	2.071	1.580	1.592	1.704	2.077	1.672	1.748	3.178	2.533	3.031	2.814	2.597	2.378	3.062	3.005	2.825
PARENet (Yao et al., 2024)	2.363	1.510	2.242	1.621	1.577	1.542	2.147	2.100	1.888	3.113	2.667	3.919	2.871	2.804	2.597	3.241	3.459	3.084
R^2	1.683	1.245	1.952	1.652	1.279	1.533	1.697	1.722	1.595	2.521	2.296	2.752	2.801	2.633	2.474	3.193	3.418	2.761
R^2 †	1.670	1.281	1.983	1.661	1.287	1.515	1.703	1.704	1.601	2.572	2.370	2.832	2.783	2.594	2.411	3.093	3.735	2.799
Translation Error(m ↓)																		
PerfectMatch (Gojcic et al., 2019)	0.059	0.070	0.079	0.065	0.074	0.062	0.093	0.065	0.071	0.082	0.098	0.096	0.101	0.080	0.089	0.158	0.120	0.103
FCGF (Choy et al., 2019)	0.053	0.056	0.071	0.062	0.061	0.055	0.082	0.090	0.066	0.084	0.097	0.076	0.101	0.084	0.077	0.144	0.140	0.100
D3Feat (Bai et al., 2020)	0.055	0.065	0.080	0.064	0.078	0.049	0.083	0.064	0.067	0.088	0.101	0.086	0.099	0.092	0.075	0.146	0.135	0.103
Predator (Huang et al., 2021)	0.048	0.055	0.070	0.073	0.060	0.065	0.080	0.063	0.064	0.081	0.080	0.084	0.099	0.096	0.077	0.101	0.130	0.093
CoFiNet (Yu et al., 2021)	0.047	0.059	0.063	0.063	0.058	0.044	0.087	0.075	0.062	0.080	0.078	0.078	0.099	0.086	0.077	0.131	0.123	0.094
GeoTrans (Qin et al., 2023)	0.047	0.053	0.079	0.057	0.061	0.051	0.081	0.079	0.064	0.071	0.089	0.089	0.091	0.090	0.063	0.117	0.106	0.090
PEAL (Yu et al., 2023)	0.045	0.050	0.080	0.059	0.062	0.056	0.077	0.069	0.062	0.069	0.082	0.089	0.094	0.088	0.056	0.109	0.094	0.085
PARENet (Yao et al., 2024)	0.043	0.050	0.080	0.058	0.060	0.048	0.078	0.075	0.061	0.069	0.087	0.090	0.097	0.093	0.056	0.109	0.122	0.090
R^2	0.041	0.042	0.071	0.059	0.047	0.050	0.065	0.069	0.056	0.067	0.078	0.079	0.097	0.087	0.060	0.112	0.100	0.085
R^2 †	0.041	0.043	0.069	0.059	0.047	0.050	0.065	0.068	0.055	0.069	0.082	0.079	0.095	0.084	0.060	0.112	0.116	0.087

C.2 Registration across Overlap Levels

Figure 7 groups all 3,005 official test pairs from 3DMatch and 3DLoMatch by metadata overlap. Non-TTA and TTA outcomes come from the identity and selected candidates of the same evaluation. Each bin reports the pooled fraction of successful pairs, rather than the scene-averaged RR used in the main table. GeoTransformer and PARE-Net are evaluated on the same pair set. TTA improves recall most in the lowest-overlap bin, while its effect is smaller and not uniformly positive at higher overlap.

C.3 Canonical Geometry and Registration Quality

Figure 8 examines the relation between predicted geometry and registration success, with and without TTA. For each pair, we average the two directional mean Euclidean nearest-neighbor distances between the predicted canonical point sets, without additional alignment. Distances are measured in normalized coordinates and are not squared. We aggregate all pairs, successful pairs, and failed pairs separately within the same overlap bins as Figure 7.

At moderate overlap, failed registrations generally have larger residuals than successful ones. Below 20% overlap, this ordering reverses for both variants: unmatched geometry contributes substantially to the full-cloud residual even when registration succeeds. TTA can therefore improve registration

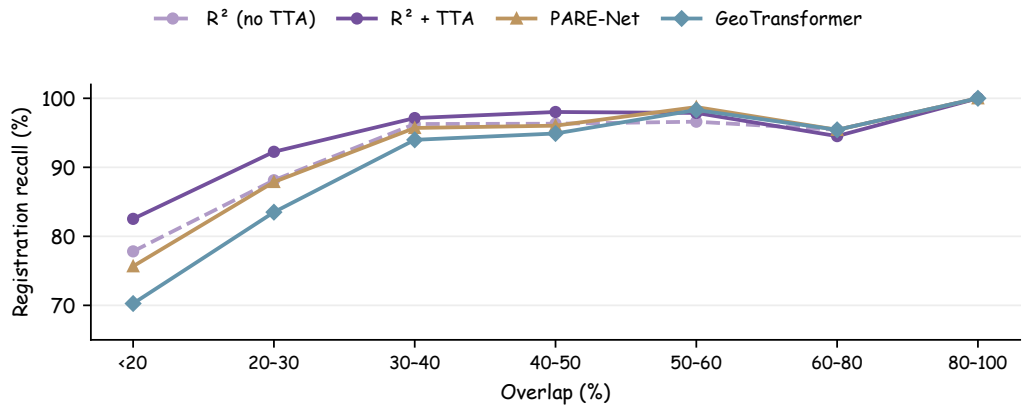

Figure 7: Registration recall by overlap on the combined indoor benchmarks. Rates pool pairs within bins; TTA uses additional forward passes.

without reducing this residual. The statistic describes cross-view proximity, but cannot by itself measure canonical prediction accuracy or certify correct alignment.

This analysis uses cached predictions with a fixed seed per pair and replays the previously selected TTA rotations. Success labels are recomputed from these same predictions. Sampling differs from the original evaluation in Table 6, so the replay is used only for this geometric analysis.

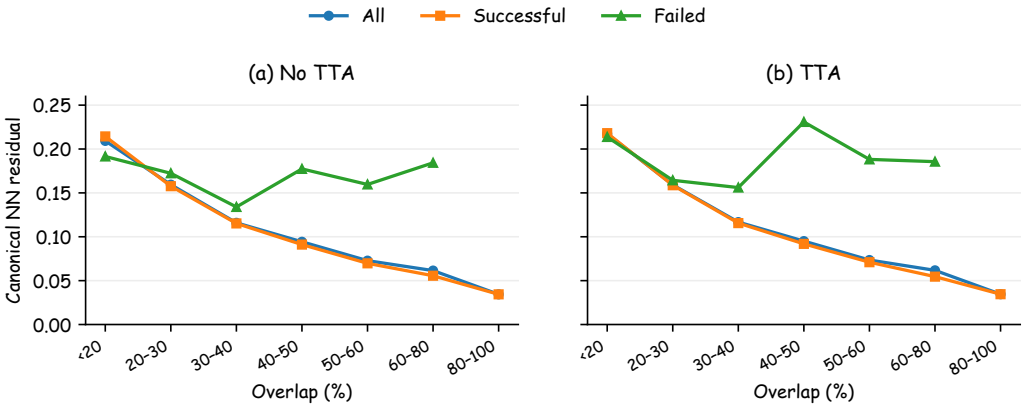

Figure 8: Canonical nearest-neighbor residual by overlap without TTA (left) and with TTA (right). Curves show mean residuals over all, successful, and failed pairs on the combined indoor benchmarks. Each variant contains all 3,005 pairs. The failed curve stops before the 80–100% bin because no failures occur there. Non-overlapping geometry affects the residual.

C.4 Qualitative Results and Failure Cases

Recovering distorted predictions with TTA. Figure 9 shows four selected 3DLoMatch failures recovered by rigidity-guided TTA. The selected canonical predictions better preserve the observed rigid structure and support accurate registration. These examples illustrate how explicit geometry provides a test-time selection signal without ground-truth poses, although a low rigidity residual alone does not guarantee correct alignment.

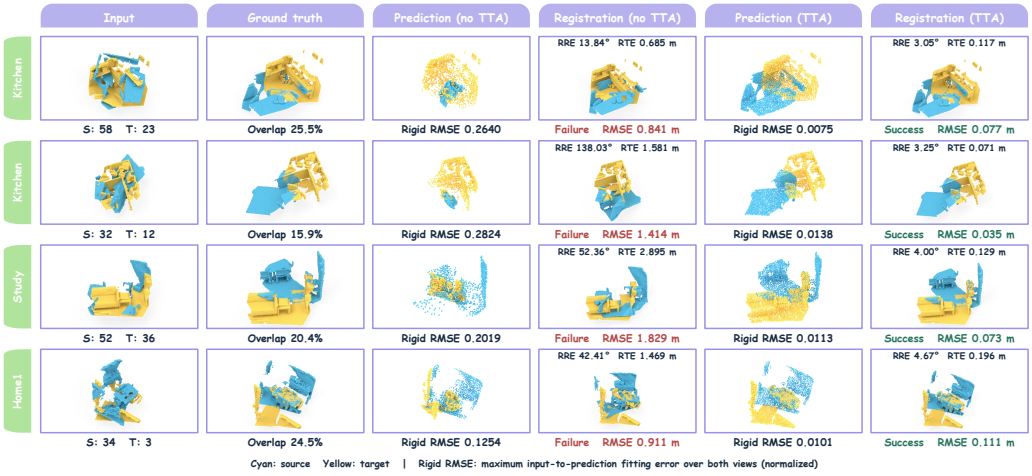

Figure 9: TTA recovery on four 3DLoMatch pairs. Each row compares the input, ground truth, and canonical predictions and registrations before and after TTA. Cyan/yellow denote source/target. Rigid RMSE is the maximum input-to-prediction rigid-fitting RMSE over both views in normalized coordinates; registration RMSE is in meters.

Inspecting the KITTI failure. Figure 10 shows the only failure among 555 KITTI test pairs. The translation error exceeds the benchmark threshold, mainly due to a vertical discrepancy. Local structures nevertheless appear better aligned under the prediction than the provided ground truth. This warrants checking point-cloud/pose consistency, but does not establish an annotation error; the pair remains counted as a failure.

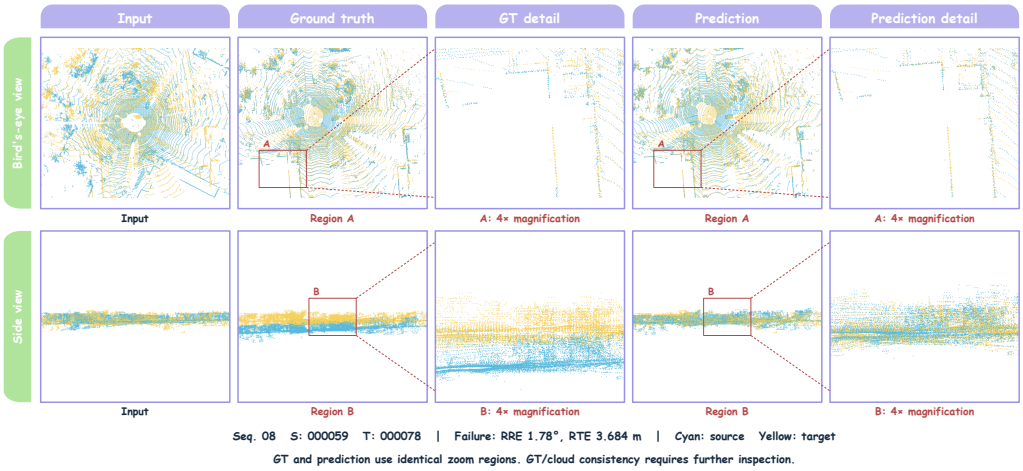

Figure 10: KITTI failure: sequence 08, source 000059 and target 000078. Top/bottom rows show bird's-eye/side views. Red boxes mark regions enlarged by $4\times$ in adjacent detail columns. Identical regions and scales are used for ground truth and prediction to compare local alignment.

D CORRESPONDENCE STABILITY AND POSE RECOVERY

This section states sufficient conditions connecting canonical prediction errors to correspondence stability and rigid pose recovery. The analysis assumes exact counterparts within the overlapping subset and does not cover completion of unobserved geometry.

D.1 CORRESPONDENCE STABILITY UNDER BOUNDED PREDICTION ERRORS

Let $\mathbf{x}_i^{s,*} \in \mathbb{R}^3$ denote the ground-truth canonical coordinate of the i-th source point, and let $\mathbf{x}_j^{t,*} \in \mathbb{R}^3$ denote the ground-truth canonical coordinate of the j-th target point. Restrict the analysis to source points in the overlapping region with exact sampled counterparts in the target view. For each such point i, assume its true corresponding target point is indexed by $\pi(i)$, such that

$$\mathbf{x}_i^{s,*} = \mathbf{x}_{\pi(i)}^{t,*}. \tag{18}$$

Let the predicted canonical coordinates be

$$\hat{\mathbf{x}}_i^s = \mathbf{x}_i^{s,*} + \mathbf{e}_i^s, \qquad \hat{\mathbf{x}}_j^t = \mathbf{x}_j^{t,*} + \mathbf{e}_j^t, \tag{19}$$

where $\mathbf{e}_i^s$ and $\mathbf{e}_j^t$ are prediction errors. Assume that these errors are uniformly bounded:

$$\|\mathbf{e}_i^s\|_2 \le \varepsilon_s, \qquad \|\mathbf{e}_j^t\|_2 \le \varepsilon_t. \tag{20}$$

Define $\varepsilon = \varepsilon_s + \varepsilon_t$. All minima over source indices below are restricted to this overlap subset; the target candidates may include non-overlapping points.

We further define the minimum cross-view separation margin as

$$\gamma = \min_i \min_{j \neq \pi(i)} \left\|\mathbf{x}_i^{s,*} - \mathbf{x}_j^{t,*}\right\|_2. \tag{21}$$

Intuitively, γ measures how far the true match is separated from all incorrect target candidates in the canonical space.

Proposition 1. If

$$\gamma > 2\varepsilon, \tag{22}$$

then the nearest-neighbor assignment in the predicted canonical space recovers the ground-truth source-to-target correspondence.

Proof. For the true match $\pi(i)$, we have

$$\left\|\hat{\mathbf{x}}_i^s - \hat{\mathbf{x}}_{\pi(i)}^t\right\|_2 \le \|\mathbf{e}_i^s\|_2 + \left\|\mathbf{e}_{\pi(i)}^t\right\|_2 \le \varepsilon. \tag{23}$$

For any incorrect target index $j \neq \pi(i)$,

$$\left\|\hat{\mathbf{x}}_i^s - \hat{\mathbf{x}}_j^t\right\|_2 \ge \left\|\mathbf{x}_i^{s,*} - \mathbf{x}_j^{t,*}\right\|_2 - \|\mathbf{e}_i^s\|_2 - \left\|\mathbf{e}_j^t\right\|_2 \tag{24}$$

$$\ge \gamma - \varepsilon. \tag{25}$$

Therefore, if $\gamma > 2\varepsilon$, then $\gamma - \varepsilon > \varepsilon$, which means every incorrect candidate is farther away than the true match. Hence the nearest-neighbor assignment is correct. □

Corollary 1. If the condition in Eq. 22 holds in both directions, every ground-truth pair in the overlapping subset is recovered as a mutual nearest-neighbor pair. This does not exclude additional mutual matches between points without observed counterparts.

This result formalizes a simple but important fact: canonical coordinate prediction error affects the stability of correspondence establishment. Once the prediction error exceeds half of the separation margin, nearest-neighbor assignments may flip, even if the downstream rigid solver itself is exact.

D.2 Pose Recovery from Correct Correspondences

After correspondence establishment in the canonical space, the rigid motion is estimated from the matched Euclidean points using a closed-form Kabsch solver. We now state the recoverability property of this step.

Let $\{(\mathbf{p}_i^s, \mathbf{p}_{\pi(i)}^t)\}_{i=1}^N$ be the matched Euclidean point pairs recovered from the canonical-space correspondences. Assume that these correspondences are correct and that the matched points are non-degenerate, i.e., they uniquely determine a 3D rigid motion.

Proposition 2. If the canonical correspondences are correct and the matched Euclidean point pairs are noise-free and non-degenerate, then the subsequent closed-form rigid alignment recovers the ground-truth transformation exactly.

Justification. This follows directly from the standard exact recovery property of the orthogonal Procrustes/Kabsch problem: when the correspondences are correct and the point configuration is non-degenerate, the unique minimizer coincides with the ground-truth rigid motion.

The above proposition shows that, under correct correspondences and non-degenerate point configurations, the final pose solver can recover the rigid transform exactly. In our pipeline, this means that pose accuracy is closely tied to whether the predicted canonical space preserves enough correspondence correctness for Kabsch-based pose recovery.

These statements give sufficient conditions under idealized assumptions. They do not establish that one canonical-frame construction is superior to another, nor do they guarantee correspondence recovery for points without observed counterparts. Practical behavior also depends on sampling, local geometric ambiguity, noise, and the conditioning of the matched points.

E Limitations and Reproducibility

E.1 Limitations and Future Work

This work studies pairwise point cloud registration and evaluates shared geometry regression on indoor RGB-D scans and KITTI LiDAR observations. When shared observations are scarce or local structures are ambiguous, the model may produce approximately rigid predictions with an incorrect relative arrangement. A low rigidity residual therefore does not guarantee correct registration. Rigidity-guided TTA can recover some difficult cases, at the cost of additional network forward passes. Future work will investigate more reliable and efficient prediction quality assessment and systematically evaluate cross-domain and cross-sensor generalization.

E.2 Code and Model Availability

The complete implementation of R^2, including training code, inference pipelines, pretrained models, and data processing utilities, will be publicly released upon acceptance.