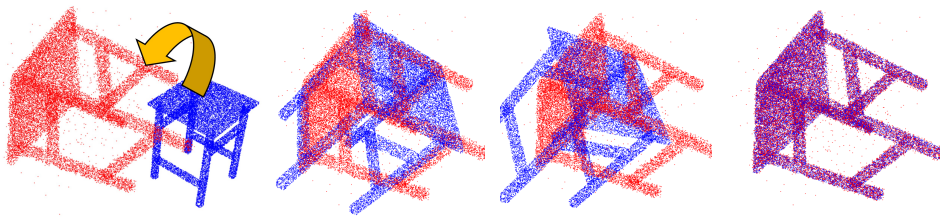
SREG: EFFECTIVELY REGISTERING POINT CLOUDS OF DIFFERENT SCALES

Anonymous authors
Paper under double-blind review



(a) Input (b) FRICP (c) FLS (d) **SReg (Ours)**

Figure 1: Illustration of full overlap point cloud registration (PCR). **(a)** input point clouds under rotation, translation, and scale differences. **(b), (c), (d)** registration results from FRICP (Zhang et al., 2021), FLS (Sun et al., 2022), and SReg (our method), respectively. The existing methods result in noticeable misalignment due to their susceptibility to local minima, while ours achieves accurate registration.

ABSTRACT

Point cloud registration (PCR) is to estimate a Sim(3) transformation, i.e., rotation, translation and scale, that aligns two point clouds. We study the challenging optimization problem of full-overlap registration for point clouds of different scales. Previous work commonly assumes the point clouds are of the same scale, and is susceptible to being trapped in a local minimum. We present SReg, a simple and training-free approach to effectively navigate the highly non-convex landscape of the problem. We propose three optimization strategies to substantially improve convergence effectiveness, namely decoupled Sim(3) transformation (to mitigate getting stuck in local minima), tangent space optimization (to overcome the discontinuities associated with large rotations), and coarse-to-fine bandwidth scheduling (to emphasize global geometric structure early on). Our extensive experiments demonstrate that SReg, compared with both SOTA learning-based and classical methods, achieves substantially lower Chamfer distance (by 69%), the highest registration recall (97%), stable registration under diverse scale and noise, and high efficiency in runtime and memory.

1 INTRODUCTION

Point cloud registration (PCR) seeks to estimate the geometric transformation that aligns two point clouds (Arun et al., 1987; Besl & McKay, 1992). Full overlap PCR is to align 3D scans that capture the same or closely similar scene, object, geometry, or environment from different perspectives or at different times (Fig. 1(a)). It is a fundamental problem in 3D computer vision, with many important applications such as 3D reconstruction and change detection (Zhang et al., 2025), human pose estimation (Baek & Gil, 2020), as well as multimodal localization and mapping (Pak et al., 2025).

Most existing methods formulate PCR problem as rigid registration, estimating only a rotation $R \in SO(3)$ and a translation $t \in \mathbb{R}^3$ (Yew & Lee, 2022; Wang & Zhao, 2017). In this work, we consider the more general and challenging Sim(3) transformation by additionally estimating an unknown scale $s \in \mathbb{R}$ for full-overlap PCR, such that the transformed source points Y take the form $e^s RY + t$.

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.

A broad range of approaches have been proposed for PCR, ranging from classical optimization-based algorithms such as ICP (Arun et al., 1987; Besl & McKay, 1992) and its robust variants (Zhang et al., 2021), to recent learning-based methods such as RegTR (Yew & Lee, 2022), GeoTransformer (Qin et al., 2022) and RAP (Pan et al., 2026). Despite substantial progress in registration accuracy and robustness, most widely adopted methods assume that the two point clouds are of the same scale and consequently estimate only a rigid transformation. This assumption is restrictive, and likely fails when point clouds are reconstructed independently, acquired using different sensing modalities, or otherwise expressed at inconsistent scales. Learning-based PCR approaches have been recently studied, which typically rely on dataset-specific training (Yew & Lee, 2022; Huang et al., 2021). While promising, they suffer from high training overhead and generalization problem for unseen scenes. One may think that for fully overlapping point clouds, a simple (and naive) approach is to preprocess the two point clouds by normalizing their scales, followed by rigid registration. However, such approach is susceptible to noise and dissimilarity between the point clouds. As we demonstrate experimentally (Sec. 5.2), it often does not recover the optimal transformation, leading to unsatisfactory performance.

PCR is notorious for its numerous local minima (Besl & McKay, 1992; Yang et al., 2016; Wang & Solomon, 2019), and the non-Euclidean structure of $SO(3)$ further complicates rotation estimation (Zhou et al., 2020). It is a formidable challenge to navigate this highly non-convex registration landscape so as to overcome being trapped at a local minimum.

We propose SReg, a novel, effective, and correspondence-free optimization algorithm for Sim(3) PCR to recover rotation, translation, and scale. SReg is training-free (does not need any dataset to train), and hence it achieves much better generalization ability to unseen point clouds. Furthermore, it converges effectively by remaining remarkably accurate despite the highly complex non-convex optimization landscape of point cloud alignment. Modeling point clouds as probability distributions, SReg achieves its convergence effectiveness with three strategies:

- *Decoupled Sim(3) Transformation:* To escape unfavorable local minima, rather than optimizing all transformation parameters jointly, SReg decouples the scale, translation, and rotation factors. This allows updates from one transformation to alter the optimization landscape seen by the others, thereby giving rise to additional opportunities.
- *Tangent Space Optimization:* To mitigate discontinuities related to the parametrization of rotations (Zhou et al., 2020), we progressively update the registration using local tangent-space approximations.
- *Coarse-to-Fine Bandwidth Scheduling:* In order to substantially reduce the tendency of rotations being trapped in local minima, SReg introduces a coarse-to-fine bandwidth scheduling strategy that takes advantage of our distribution-based approach to emphasize large-scale geometric structures early on, and progressively incorporates finer geometric details.

We conduct extensive experiments, and show in our ablation studies (Sec. 5.4) that our optimization strategies greatly improve convergence effectiveness. We compare SReg against SOTA learning-based methods, classical registration methods, and rigid-registration baselines. In Fig. 1, we qualitatively compare SReg with SOTA methods. Clearly, ours achieves much better alignment. Our experimental results further show that SReg achieves substantially lower Chamfer distance (by 69%) and the highest registration recall (97%), outperforming the other methods even when they employ scale-normalization strategies. Moreover, our method remains robust across substantial scale and noise variations, maintaining over 91% registration recall for scale differences as large as $2.5\times$, low Chamfer distances (below 0.0070) even under high Gaussian jitter *and* additional outlier points, and high efficiency in runtime and memory.

2 Related Works

Existing PCR approaches can be broadly divided into two categories: (1) classical geometry and optimization-based methods, which directly minimize the distance between corresponding points; and (2) learning-based methods, which employ learned representations or correspondence predictors to facilitate alignment.

2.1 CLASSICAL METHODS

The Iterative Closest Point (ICP) algorithm of Besl & McKay (1992) is one of the first PCR methods by alternating between nearest-neighbor correspondence estimation and rigid transformation optimization. Closely related point-to-plane formulations (Chen & Medioni, 1992) improves convergence by exploiting local surface geometry. Despite their simplicity, efficiency, and training-free nature, ICP-based methods are inherently local optimizers and are therefore sensitive to local minima. Subsequent works have sought to improve these limitations from different perspectives. Jian & Vemuri (2011) introduced a Gaussian-mixture approach to rigid registration. Sparse ICP (Bouaziz et al., 2013) introduced sparsity-optimized norms to increase robustness to outliers, while Go-ICP (Yang et al., 2016) combined ICP with branch-and-bound search to provide globally optimal rigid registration. Fast and Robust ICP (Zhang et al., 2021) combines robust error functions and Anderson acceleration to improve both robustness and convergence speed.

Although the vast majority of classical registration methods assume a common scale, a small number of works have explicitly considered scale estimation. SAICP (Sahillioğlu & Kavan, 2021) extends the ICP framework by incorporating scale into its iterative closest-point optimization, while FLS (Sun et al., 2022) represents point clouds as continuous functions in an orthonormal basis and estimates transformations by minimizing functional misalignment. These works demonstrate that scale-aware PCR is feasible, but they address the problem within substantially different optimization approaches. SAICP remains based on iteratively estimated closest-point correspondences, whereas FLS represents point clouds using orthonormal basis functions and focuses on a functional least-squares.

In contrast, our work investigates direct Sim(3) registration using a distribution-based approach, with the central challenge being how to reliably optimize over the highly non-convex landscape. Our method addresses this gap through three complementary strategies: decoupling transformations (Sec. 4.1), performing local tangent space updates (Sec. 4.2), and progressively refining rotations through coarse-to-fine bandwidth scheduling (Sec. 4.3). Thus, our contribution is not simply the inclusion of scale, but a training-free distribution-based method for directly recovering Sim(3) transformations, while explicitly addressing the optimization difficulties that arise in direct registration.

2.2 LEARNING-BASED METHODS

With the success of deep learning on 3D data, recent registration methods have increasingly replaced hand-crafted geometric representations with learned features and correspondence estimation. Early approaches such as PointNetLK (Aoki et al., 2019) combined PointNet (Qi et al., 2017a;b) feature representations with an iterative feature-alignment algorithm to estimate rigid transformations between point clouds, while Deep Closest Point (DCP) (Wang & Solomon, 2019) employs learned point embeddings, attention, and differentiable SVD to estimate rigid transformations. The work by Yew & Lee (2020) subsequently introduces learned hybrid features and differentiable soft correspondence assignments in RPM-Net, improving robustness to initialization, noise, and partial visibility. Later works increasingly focused on the particularly challenging low-overlap registration, such as with PREDATOR (Huang et al., 2021), PointDSC (Bai et al., 2021), RegTR (Yew & Lee, 2022) and GeoTransformer (Qin et al., 2023).

These learning-based approaches have demonstrated impressive robustness, especially when reliable geometric correspondences are difficult to establish. However, the target goal remains the same: estimating rotation and translation under an assumed common scale. Learned models also require a training procedure and may not generalize well to unseen data without a sacrifice in performance. By contrast, our method requires no training and works directly out-of-the-box. In our experiments, we show that our training-free algorithm correctly aligns substantially more pairs of point clouds than pretrained models (Sec. 5.2), even with no prior knowledge of the dataset. This shows the effectiveness of SReg in generalizing to unseen point clouds.

3 Optimization Problem: Formulation and Distribution Matching Approach

Let $X = \{x_k\}_{k=1}^n$ (the target) and $Y = \{y_k\}_{k=1}^m$ (the source) be two point clouds representing the same object in $\mathbb{R}^3$, and hence X and Y have full or near-full overlap. The goal of PCR is to find the required rotation $R \in SO(3)$, translation $t \in \mathbb{R}^3$, and scale $s \in \mathbb{R}$ such that the source point cloud moves to align with the target point cloud, i.e.,

$$e^s RY + t = X. \tag{1}$$

Prior approaches based on explicit point-to-point correspondence estimation struggle in the cases with large rotations, translations and scale differences. We hence attack the PCR problem from a distribution-matching perspective. Specifically, if we consider the points in a point cloud as samples generated from some hidden underlying distribution, an accurate transformation should then cause the two aligned distributions to closely agree. Mathematically, we can estimate the distribution of X using Gaussian kernel density estimation (KDE),

$$\text{Distribution of } X\text{: } f(t) = \frac{1}{n}\sum_{k=1}^{n} \frac{1}{(2\pi\sigma_x^2)^{3/2}} \exp\left(-\frac{\|t - x_k\|^2}{2\sigma_x^2}\right), \tag{2}$$

where σ_x is the kernel bandwidth chosen using Silverman's rule of thumb (Silverman, 1986). Similarly, we can apply KDE to estimate the distribution $g(t)$ of Y. The "distance" between the two point clouds can then be expressed through the $\mathcal{L}^2$ distance between the KDEs,

$$\|f - g\|_2^2 = \int_{\mathbb{R}^3} |f(t) - g(t)|^2 \, dt. \tag{3}$$

Since the product of Gaussian kernels can be evaluated in closed-form (Rasmussen & Williams, 2005), this distance can be minimized directly using gradient descent without discretizing the underlying density or resorting to numerical integration. However, as we found during testing, directly optimizing this over the full Sim(3) transformation is difficult due to the highly non-convex landscape of PCR. We thus introduce three optimization strategies (Sec. 4) to address this issue.

4 Training-Free Optimization Strategies

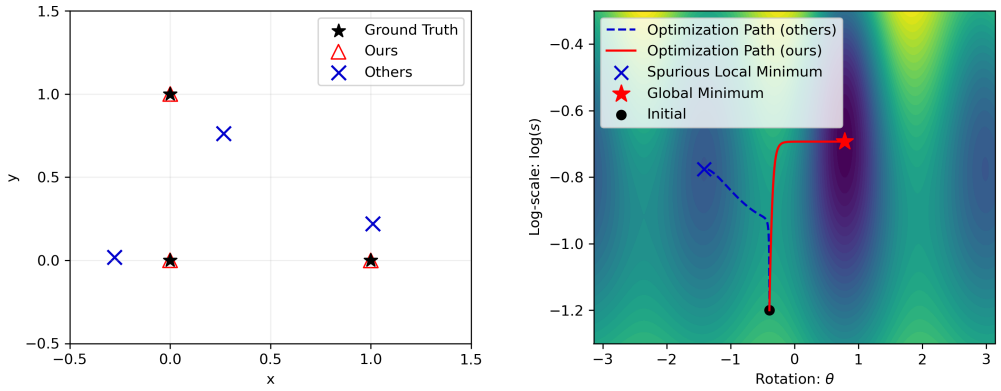
In this section, we present the three complementary optimization strategies to improve convergence effectiveness in the highly non-convex registration landscape: 1) decoupled Sim(3) transformation, where we decouple translation, scale, and rotation into separate transformation-specific objectives (Sec. 4.1); 2) tangent space optimization, where we estimate the transformation through a sequence of local incremental updates rather than a single global optimization (Sec. 4.2); and 3) coarse-to-fine bandwidth scheduling, where we emphasize global structure early during optimization and progressively introduce finer geometric details (Sec. 4.3). We conclude with the full algorithm integrating the three strategies in Sec. 4.4.

4.1 Decoupled Sim(3) Transformation

Rather than representing each point cloud using a single probability density and jointly optimizing all transformations, we construct separate, transformation-specific density representations for translation, scale, and rotation, and compute the total loss as the sum of the three individual losses,

$$\mathcal{L} = \mathcal{L}_t + \mathcal{L}_s + \mathcal{L}_r. \tag{4}$$

Each representation is chosen to isolate the geometric effect of the corresponding transformation while suppressing as much as possible the influence of the others. Specifically, for each point cloud, KDE is used to estimate 3 translational densities for each of the x, y, z coordinates, 1 scale density for the distribution of the *radial distances* to the centroid of the point cloud, and 1 rotational density for the three-dimensional distribution of the *centered* point cloud. This decomposition gives rise to separate objectives for the three transformation components. Appendix C provides the exact formulas for each density, and Appendix D provides the fully expanded loss functions.



(a) Registration results

(b) Optimization landscape

Figure 2: A 2D example to illustrate the common issue of local minima in PCR, where only a scaling factor s and a rotational angle θ are estimated. *Others* refers to joint optimization, while *Ours* refers to our decoupled transformations. **Left:** Our decoupled strategy converges correctly to the ground truth (global minimum), while joint optimization gets stuck in an inaccurate local minimum. **Right:** Despite the same initialization, our decoupled strategy isolates the effects of each transformation and reaches the global minimum, while joint optimization becomes trapped in a local minimum. Tests on real 3D point clouds show similar behavior (Sec. 5.4).

We design each density to represent projections of the point cloud onto lower dimensional spaces, because (1) functions on lower dimensional spaces can be easier to learn (Mousavi-Hosseini et al., 2023), (2) it allows us to decouple the search space into invariant subspaces, and (3) it *greatly* reduces the size of the search space when one or more of the position, scale or rotation are already accurate to begin with. Notice that by centering the point clouds before computing the rotational density, we isolate the effect translation might have on registration. In the same spirit, by considering only the radial distances when computing the scale density, we isolate the effect of the point cloud's position and rotation. Thus, splitting up the features allows us to selectively optimize for the transformations needed. Furthermore, it allows us to easily escape many of the local minima in the highly non-convex search space, since a local minimum with respect to a single, lower-dimensional objective may not be a local minimum for the other objectives.

The simplified 2D example in Figure 2 gives an intuitive understanding of the common local minima issue in PCR. In this example, we create a simple point cloud in 2D consisting of only 3 points (Fig. 2a), then augment it with scale and rotation only. Joint optimization becomes trapped in a spurious local minimum, producing an incorrect solution despite starting from the same initialization. On the other hand, our decoupled objectives are designed to isolate the effects of each transformation, allowing the correct scale and rotation to be recovered without interference from the other variable (Fig. 2b). Tests on real 3D point clouds show similar behavior (Sec. 5.4).

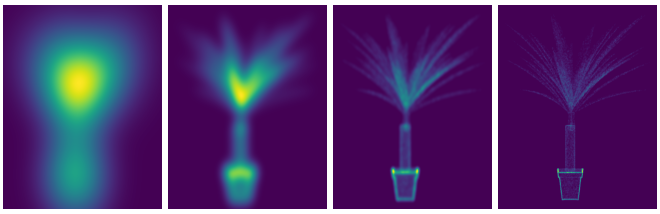
4.2 Tangent Space Optimization

Directly optimizing 3D rotations through Euclidean parameterizations is particularly challenging (Zhou et al., 2020). During our testing, we found that doing so results in suboptimal performance, especially when large rotational updates are required.

To solve this, we update rotations as follows. We start with a unit quaternion $dq = (1, 0, 0, 0)$ corresponding to the identity rotation, and perform gradient descent to obtain a local rotational update. The optimized quaternion is normalized and converted into a rotational matrix R. The current rotation estimate R_k is then updated by composing it with this incremental rotation R,

$$R_{k+1} = RR_k, \tag{5}$$

after which the source point cloud is transformed using this incremental rotation. Before the start of the next iteration, the quaternion parameters are reset to $dq = (1, 0, 0, 0)$ and the procedure is



(a) $64\sigma_R$ (b) $16\sigma_R$ (c) $4\sigma_R$ (d) σ_R

Figure 3: Coarse-to-fine KDE representations under decreasing bandwidth. Larger bandwidths emphasize global geometric structure, ensuring the overall orientation is correct. As the bandwidth decreases, finer geometric details are revealed, allowing for more precise alignment.

repeated. In this way, a potentially large global rotation is recovered through a sequence of small local rotational updates, avoiding the need to find a single smooth function that globally parametrizes $SO(3)$. For simplicity, we also use the same tangent-space optimization for translation and scale.

4.3 Coarse-to-Fine Bandwidth Scheduling

Estimating probability densities using KDE requires the choice of a bandwidth parameter σ. This choice involves an *inherent tradeoff*: A larger bandwidth gives us a smoother density function at the cost of suppressing geometric details, while a smaller bandwidth reveals more fine-grained structure that may not be necessary during the early stages of optimization (Fig. 3).

Rather than choosing a fixed bandwidth and compromising between these two properties, we *embrace* this tradeoff to further boost the accuracy of rotations. We begin with a larger σ to guide the rotation towards a correct global alignment. Then, we progressively reduce σ as optimization continues, gradually introducing finer details and enabling precise alignment once a reasonable coarse alignment has been obtained. Mathematically, we use the following schedule to determine the bandwidth at every iteration k,

$$\sigma_r^{(k)} = \sigma_R \cdot \gamma^{\max(0,1-2k/K)}, \tag{6}$$

where K is the total number of optimization iterations, $\gamma > 1$ controls the initial degree of smoothing, and σ_R denotes the bandwidth selected using Silverman's rule of thumb (Silverman, 1986). In our implementation, we use $\gamma = 16$.

4.4 The Complete Algorithm

We now summarize the complete registration procedure in Algorithm 1. Given a target point cloud X and a source point cloud Y, we first estimate the transformation-specific representations introduced in Section 4.1. KDE bandwidths are then estimated from the target point cloud and kept fixed throughout optimization, except for the rotational bandwidth, which follows the coarse-to-fine schedule described in Section 4.3.

Registration proceeds iteratively through a sequence of local Sim(3) transformations. At each iteration, we initialize the parameters dt, ds, dq to identity, then differentiate $\mathcal{L} := \mathcal{L}_t + \mathcal{L}_s + \mathcal{L}_r$ with respect to dt, ds, dq to obtain a small, local transformation dH (Section 4.2). We use dH to update Y, and recompute the transformation-specific representations. The procedure is then repeated. In this way, we recover the final alignment through a sequence of small local updates.

5 Illustrative Experimental Results

We evaluate SReg on ModelNet40 (Wu et al., 2015) under both rigid (translation and rotation only) and Sim(3) augmentations (translation, rotation and scale) (Sec. 5.2). Then we show the robustness of SReg against large differences in scale (Sec. 5.3) and measurement noise in point clouds (Sec. 5.3). Finally, we present ablation studies on our optimization strategies (Sec. 5.4).

Algorithm 1 SReg

Require: Target point cloud X, source point cloud Y, number of iterations K
$H \leftarrow$ Identity matrix ▷ H accumulates the overall Sim(3) transformation
Estimate KDE bandwidths σ_t, σ_s, and σ_r
$f_t, f_s, f_r \leftarrow$ Translational, scale and rotational densities of X
for $k = 1, \cdots, K$ **do**
  $g_t, g_s, g_r \leftarrow$ Translational, scale and rotational densities based on current Y
  $dt, ds, dq \leftarrow (0,0,0),\ 0,\ (1,0,0,0)$ ▷ Initialize parameters
  Set rotation bandwidth $\sigma_r^{(k)}$ using the coarse-to-fine schedule ▷ Sec. 4.3
  $\mathcal{L} \leftarrow \mathcal{L}_t(f_t, g_t, dt) + \mathcal{L}_s(f_s, g_s, ds) + \mathcal{L}_r(f_r, g_r, dq)$ ▷ Sec. 4.1
  Differentiate $\mathcal{L}$ with respect to dt, ds, and dq
  Step optimizer to update dt, ds, and dq ▷ dt, ds, dq now contains a small transformation
  Construct dH from dt, ds, and dq
  Update source point cloud Y using dH
  $H \leftarrow dH \cdot H$ ▷ Accumulate overall transformation in H
end for
return H

5.1 Experimental Setup and Evaluation Metrics

Data augmentation. We conduct all experiments on ModelNet40 (Wu et al., 2015), a widely used benchmark containing 12,311 CAD models from 40 object categories. Following Yew & Lee (2022; 2020), we use the last 20 categories for testing. For each CAD model, we sample 2048 points from the mesh surface, and normalize each point cloud to fit within the unit sphere. To generate registration pairs, we follow the aforementioned methods by applying random rotations, translations and jittering to one of the point clouds. In addition, we add 2% extra $\mathcal{N}(0, 0.3^2)$ Gaussian outliers to one of the point clouds. For experiments involving Sim(3) registration, we apply a random scale factor e^s to one point cloud, where $s \sim \mathcal{U}[\log(1/1.5), \log(1.5)]$.

Comparison methods. The PCR literature contains a large number of methods with substantially different considerations and architectures. Rather than attempting an exhaustive comparison, we select representative approaches from the two paradigms (classical, learning-based) discussed in Section 2. For classical methods, we evaluate against FRICP (Zhang et al., 2021). For learning-based methods, we evaluate against PREDATOR (Huang et al., 2021), RegTR (Yew & Lee, 2022) and RAP (Pan et al., 2026). While most PCR methods assume a common scale, we were able to identify SAICP (Sahillioğlu & Kavan, 2021) and FLS (Sun et al., 2022) as particularly relevant methods (training-free, explicitly estimates scale), making them the closest in settings to our method. We use the released implementations and pretrained models of the respective methods where applicable.

Evaluation metrics. Let $(\hat{s}, \hat{R}, \hat{t})$ and $(s_{\text{gt}}, R_{\text{gt}}, t_{\text{gt}})$ denote the estimated and ground truth transformations respectively. Following the standard evaluation metrics in PCR works (Wang & Solomon, 2019), we report the relative rotation error (RRE) and relative translation error (RTE). For experiments involving unknown scale, we additionally report the relative log-scale error (RLSE). These are defined as

$$\text{RRE} = \frac{180}{\pi} \arccos\left(\frac{\text{tr}(\hat{R}^T R_{\text{gt}}) - 1}{2}\right), \quad \text{RTE} = \|\hat{t} - t_{\text{gt}}\|_2, \quad \text{RLSE} = \left|\log \frac{\hat{s}}{s_{\text{gt}}}\right|. \tag{7}$$

We further show the modified squared Chamfer distance (CD) used by RegTR (Yew & Lee, 2022), as well as the registration recall (RR), defined as the percentage of registration pairs satisfying all of RRE $< 1^\circ$, RTE < 0.01, and CD < 0.01. For experiments involving Sim(3) registration, we additionally add the condition RLSE < 0.05, and label this metric as RR_s. Higher RR, RR_s, and lower RRE, RTE, RLSE, and CD indicate better performance. Besides accuracy, we have also benchmarked the runtime and peak GPU memory consumption in Appendix A.

Implementation details. Our method requires no pretraining. For each point-cloud pair, we perform 1,000 optimization iterations with a fixed learning rate of 0.005.

Table 1: Rigid registration results. Although our method does not achieve the lowest RTE or CD, it successfully registers a substantial 97.87% of pairs, demonstrating its effectiveness.

Method	RR (%) ↑	RRE (°) ↓	RTE ↓	CD ↓
FRICP	89.73	5.8595	0.0054	0.0032
FLS	31.60	7.4922	0.0055	0.0037
SAICP	91.00	3.6482	**0.0039**	0.0025
PREDATOR	47.47	1.5206	0.0082	0.0016
RegTR	77.80	1.3019	0.0060	**0.0015**
RAP	81.28	16.7273	0.0449	0.0267
SReg (Ours)	**97.87**	**0.5648**	0.0091	0.0025

Table 2: Sim(3) registration results. *ScaleNorm* denotes scale normalization is applied prior to registration. Despite giving other methods this *ScaleNorm* advantage, our method *still* outperforms them, achieving the highest RR of 97.16%.

Method	RR_s (%) ↑	RRE (°) ↓	RTE ↓	RLSE ↓	CD ↓
ScaleNorm + FRICP	69.12	5.8254	0.0175	0.0377	0.0046
FLS	27.65	9.1072	0.0149	**0.0048**	0.0124
SAICP	91.23	3.5858	**0.0042**	0.0145	0.0032
ScaleNorm + PREDATOR	33.41	2.2893	0.0247	0.0377	0.0037
ScaleNorm + RegTR	58.29	1.4592	0.0146	0.0377	0.0030
ScaleNorm + RAP	63.11	17.2785	0.0537	0.0377	0.0273
SReg (Ours)	**97.16**	**0.5738**	0.0093	0.0067	**0.0028**

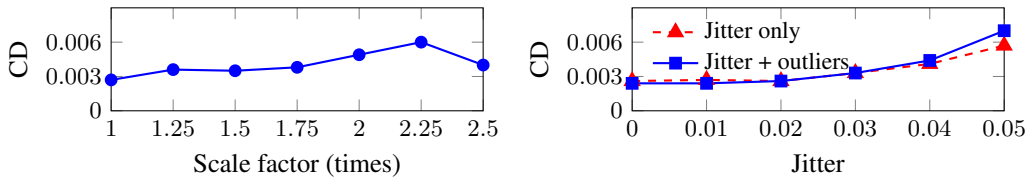
5.2 Results on ModelNet40

We first evaluate registration without scale augmentation. This experiment tests whether SReg remains effective in the rigid setting. Table 1 shows that our method achieves the highest RR of 97.87%, exceeding the strongest other method, SAICP, by 6.87 percentage points. It also obtains the lowest RRE of $0.5648°$. Although we do not achieve the lowest RTE or CD, our method still aligns a *much* larger fraction (97.87%) of pairs successfully, demonstrating its effectiveness even without a scale mismatch.

Next, we introduce scale augmentation to the registration problem. For methods that do not inherently estimate scale, we provide a strong scale-normalization baseline by independently re-scaling each point cloud to fit within a unit sphere before registration. We apply this procedure to FRICP (Zhang et al., 2021), PREDATOR (Huang et al., 2021), RegTR (Yew & Lee, 2022) and RAP (Pan et al., 2026), and denote this by writing "*ScaleNorm* + Method" in Table 2. For methods that directly estimate scale (SAICP, FLS, ours), we do not give this advantage. Table 2 shows the results. Qualitative results are shown in Appendix B. Note that since we augment scale by randomly enlarging or shrinking the point clouds, scale normalization essentially *reverses the applied augmentation*, up to point jittering and Gaussian noise factors. As such, scale normalization provides a *substantial* advantage to those methods by effectively converting the Sim(3) registration problem into a rigid registration problem. Despite these advantages, our method *still* achieves the highest 97.16% registration recall, compared with 91.23% for SAICP and 69.12% for the strongest scale-normalized rigid baseline, FRICP. Our method also obtains the lowest RRE of $0.5738°$ and CD of 0.0028, respectively. These results demonstrate the benefit of directly estimating a Sim(3) transformation over scale-normalization extensions to rigid registration methods.

5.3 Sensitivity to Scale, Noise and Outliers

We further evaluate the robustness of our method to scale. Specifically, we apply a fixed scale instead of random scaling. We only consider scale factors greater than 1, as exchanging the source and target point clouds replaces a scale factor s with its reciprocal $1/s$. All other experimental



(a) CD under increasing scale factors.

(b) CD under increasing jitter.

Figure 4: Sensitivity of our method to scale variation and measurement noise.

Table 3: Ablation studies conducted on challenging transformations. It shows the importance of each optimization strategy: removing even one *greatly* reduces registration performance.

Method	RR_s (%) ↑	RRE (°) ↓	RTE ↓	RLSE ↓	CD ↓
Full method	**89.89**	**8.9256**	**0.0085**	**0.0068**	**0.0048**
w/o Decoupling, Sec. 4.1	76.22	21.4272	0.0257	0.0148	0.2630
w/o Tangent Space Updates, Sec. 4.2	0.00	30.2015	0.3812	**0.0068**	0.1258
w/o Coarse-to-Fine, Sec. 4.3	56.48	38.0779	0.0399	**0.0068**	0.0355

settings are unchanged. Our method remains stable across the tested range (Fig. 4a), maintaining low Chamfer distances even under scale discrepancies as large as $2.5\times$.

We next evaluate the robustness of our method to noise and additional outliers. Compared with the previous experiments where we followed Yew & Lee (2022) and used $\mathcal{N}(0, 0.01^2)$ Gaussian jitter, we now consider a wider range of noise levels under two settings: (1) with positional jitter applied only (dashed, triangle mark in Figure 4b), and (2) with positional jitter *and* extra outlier points (solid, square mark in Figure 4b). All other experimental settings remain unchanged. Figure 4b shows that Chamfer distance remains bounded, even under as large as $5\times$ noise.

5.4 Ablation Studies

We conduct ablation studies to evaluate the contribution of each optimization strategy (Sec. 4). Unlike previous experiments where transformations were randomly sampled, we deliberately fix the rotation, translation, and scale transformations to their respective maximum values. We adopt this more challenging setting because registration becomes increasingly difficult as the transformation magnitudes grow, making failures caused by unfavorable local minima substantially more apparent. As shown in Table 3, removing any of the three strategies greatly impacts registration performance.

6 Conclusion and Future Work

We have presented SReg, an effective training-free optimization algorithm for Sim(3) point cloud registration. Through the use of decoupled transformations, tangent space optimizations, and coarse-to-fine bandwidth scheduling, we have shown that SReg converges effectively, attaining remarkably high registration accuracy despite the complex non-convex PCR landscape. Our extensive experiments on ModelNet40 demonstrate that SReg achieves the highest registration recall of 97% and substantially lower (by 69%) Chamfer distance compared with competing SOTA methods, while maintaining high robustness to scale and measurement noise, and high efficiency in runtime and memory.

Like many other prior works, we have considered point clouds with full or near-full overlap. While such case is often observed in practice, we are currently extending our study to the more general case of partially overlapping point clouds. Using our correspondence-free, distribution-based method, we are investigating how to identify overlapping areas in a robust and efficient manner.

AI USE STATEMENT

In this work, we used generative AI tools to assist with coding, simplifying mathematical expressions, providing feedback on research methodology and experiments, manuscript drafting and literature review. We did not use generative AI to clean and reformat datasets, run code or run experiments. All AI-assisted mathematical simplifications, code, literature references, and written content were manually reviewed and verified by the authors, and all reported experiments were conducted by the authors. We take full responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code and data used to reproduce our results in this paper are available in the supplementary materials. Detailed derivations of the formulas used are provided in the appendix.

REFERENCES

Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust and efficient point cloud registration using pointnet. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987. doi: 10.1109/TPAMI.1987.4767965.

Seongmin Baek and Younhee Gil. Human pose estimation using articulated icp. In *Proceedings of the 2019 2nd International Conference on Control and Robot Technology*, ICCRT 2019, pp. 125–129, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450372527. doi: 10.1145/3387304.3387309. URL https://doi.org/10.1145/3387304.3387309.

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, 2021. URL https://arxiv.org/abs/2103.05465.

P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. doi: 10.1109/34.121791.

Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse iterative closest point. *Computer Graphics Forum*, 32(5):113–123, 2013. doi: https://doi.org/10.1111/cgf.12178. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12178.

Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992. ISSN 0262-8856. doi: https://doi.org/10.1016/0262-8856(92)90066-C. URL https://www.sciencedirect.com/science/article/pii/026288569290066C. Range Image Understanding.

Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap, 2021. URL https://arxiv.org/abs/2011.13005.

Bing Jian and Baba C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, 2011. doi: 10.1109/TPAMI.2010.223.

Alireza Mousavi-Hosseini, Sejun Park, Manuela Girotti, Ioannis Mitliagkas, and Murat A. Erdogdu. Neural networks efficiently learn low-dimensional representations with sgd, 2023. URL https://arxiv.org/abs/2209.14863.

Gyuhyeon Pak, Hae Min Cho, and Euntai Kim. G2s-icp slam: Geometry-aware gaussian splatting icp slam, 2025. URL https://arxiv.org/abs/2507.18344.

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 *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2026.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017a. URL https://arxiv.org/abs/1612.00593.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017b. URL https://arxiv.org/abs/1706.02413.

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 (CVPR)*, pp. 11143–11152, June 2022.

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, 2023. URL https://arxiv.org/abs/2308.03768.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

Yusuf Sahillioğlu and Ladislav Kavan. Scale-adaptive icp. *Graphical Models*, 116:101113, 2021. ISSN 1524-0703. doi: https://doi.org/10.1016/j.gmod.2021.101113. URL https://www.sciencedirect.com/science/article/pii/S1524070321000187.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

Muchen Sun, Allison Pinosky, Ian Abraham, and Todd Murphey. Scale-invariant fast functional registration, 2022. URL https://arxiv.org/abs/2209.12763.

Fang Wang and Zijian Zhao. A survey of iterative closest point algorithm. In *2017 Chinese Automation Congress (CAC)*, pp. 4395–4399, 2017. doi: 10.1109/CAC.2017.8243553.

Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration, 2019. URL https://arxiv.org/abs/1905.03304.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes, 2015. URL https://arxiv.org/abs/1406.5670.

Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, November 2016. ISSN 2160-9292. doi: 10.1109/tpami.2015.2513405. URL http://dx.doi.org/10.1109/TPAMI.2015.2513405.

Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11821–11830. IEEE, June 2020. doi: 10.1109/cvpr42600.2020.01184. URL http://dx.doi.org/10.1109/CVPR42600.2020.01184.

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

Chenhao Zhang, Yezhi Shen, and Fengqing Zhu. Icp-3dgs: Sfm-free 3d gaussian splatting for large-scale unbounded scenes, 2025. URL https://arxiv.org/abs/2506.21629.

Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. ISSN 1939-3539. doi: 10.1109/tpami.2021.3054619. URL http://dx.doi.org/10.1109/TPAMI.2021.3054619.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks, 2020. URL https://arxiv.org/abs/1812.07035.

APPENDIX

A RUNTIME AND GPU MEMORY USAGE

We report the average wall-clock runtime required for one ModelNet40 registration. GPU-enabled methods are evaluated using a single NVIDIA RTX 4090 GPU, while CPU-only methods use their released CPU implementations. Table 4 shows the results. Note that PREDATOR, RegTR and RAP are feed-forward learning-based methods and are therefore expected to achieve lower inference times compared to optimization-based approaches.

We additionally measure GPU memory usage by recording the peak memory consumption of each registration in the ModelNet40 dataset. Methods that do not use GPU acceleration are marked as *CPU only*. Table 4 shows the results.

Table 4: Average runtime and peak GPU memory per ModelNet40 registration. GPU-enabled methods are evaluated using a single NVIDIA RTX 4090 GPU. PREDATOR, RegTR and RAP are feed-forward learning-based methods and are therefore expected to achieve lower inference times compared to optimization-based approaches.

Method	Paradigm	Runtime (s) ↓	GPU Memory (GB) ↓
FRICP	Optimization-based	0.7536	CPU only
FLS	Optimization-based	4.4666	1.7759
SAICP	Optimization-based	2.1326	CPU only
PREDATOR	Learning-based	0.0565	0.3303
RegTR	Learning-based	0.0267	0.4260
RAP	Learning-based	0.3005	1.4995
SReg (Ours)	Optimization-based	2.6091	0.3821

B QUALITATIVE RESULTS

We illustrate in Table 5 some examples extracted from the experimental results presented in Sec. 5.2. Each row in the table shows the same registration example across the ground truth and evaluated methods.

C KDE DENSITY FORMULAS

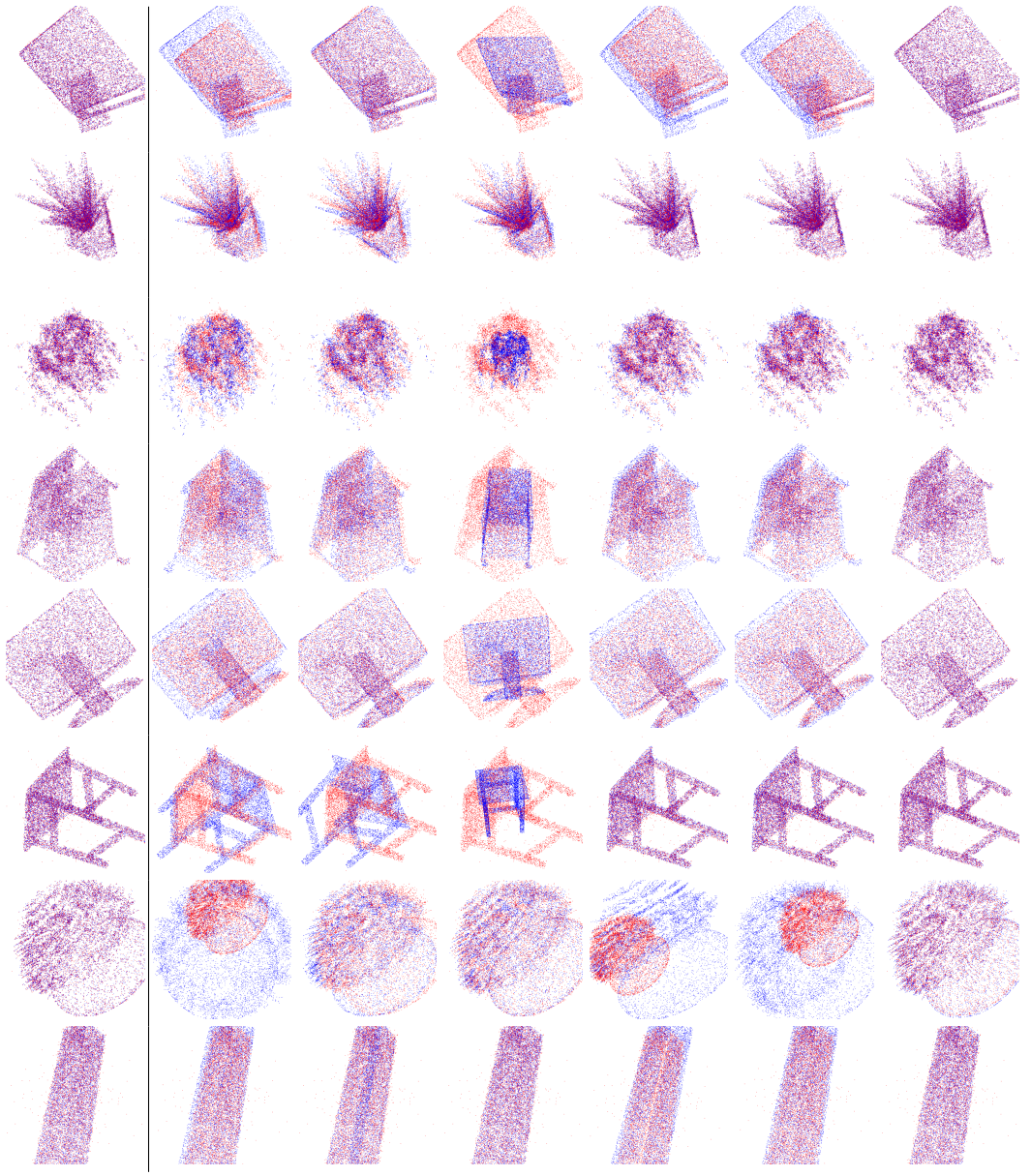
In this section, we derive the translational, scale and rotational KDE density formulas (Sec. 4.1). We first define the notations that will be used, then write out their formulas.

Notation. Let

- the target and source point clouds be denoted respectively by
$$X = \{x_k\}_{k=1}^n, \qquad Y = \{y_k\}_{k=1}^m, \qquad x_k, y_k \in \mathbb{R}^3, \tag{8}$$
- the centroids of the two point clouds be denoted by
$$\bar{x} = \frac{1}{n}\sum_{k=1}^{n} x_k, \qquad \bar{y} = \frac{1}{m}\sum_{k=1}^{m} y_k, \tag{9}$$
- the average radial distances of the two point clouds be denoted by
$$\|X\| = \frac{1}{n}\sum_{k=1}^{n} \|x_k - \bar{x}\|, \qquad \|Y\| = \frac{1}{m}\sum_{k=1}^{m} \|y_k - \bar{y}\|, \tag{10}$$

Table 5: Qualitative registration results on ModelNet40. Each row shows the same registration example across the ground truth and evaluated methods.

Ground Truth	FRICP	FLS	SAICP	PREDATOR	RegTR	**SReg (Ours)**



- the standard basis in $\mathbb{R}^3$ be denoted by e_1, e_2, e_3,
- the learnable incremental translation be denoted by $dt = (dt_1, dt_2, dt_3) \in \mathbb{R}^3$,
- the learnable logarithmic scale be denoted by $ds \in \mathbb{R}$, such that the corresponding positive scale factor is e^{ds},
- the learnable rotation parameter be denoted by $dq \in \mathbb{R}^4$, with $R_{dq} \in SO(3)$ denoting its corresponding rotation matrix, and
- the KDE bandwidths used for translation, scale, and rotation be denoted by σ_t, σ_s and σ_r, respectively.

Translational KDE. For each coordinate axis $i \in \{1, 2, 3\}$, we construct a one-dimensional KDE using the projection of each point onto e_i. For point cloud X, we have

$$f_i(t) := \frac{1}{n} \sum_{k=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-\frac{(t - \langle e_i, x_k \rangle)^2}{2\sigma_t^2} \right). \tag{11}$$

For point cloud Y, the learnable translation dt_i shifts the KDE centers along the i-th coordinate axis, i.e.,

$$g_i(t; dt_i) := \frac{1}{m} \sum_{k=1}^{m} \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-\frac{(t - \langle e_i, y_k \rangle - dt_i)^2}{2\sigma_t^2} \right). \tag{12}$$

Scale KDE. To estimate scale independently of translation and rotation, we use the radial distance of every point from the centroid of its corresponding point cloud. The radial distributions are additionally centered by their respective mean radial distances.

The one-dimensional scale KDE for point cloud X is then

$$f_s(t) := \frac{1}{n} \sum_{k=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_s} \exp\left[-\frac{(t - [\|x_k - \bar{x}\| - \|X\|])^2}{2\sigma_s^2} \right]. \tag{13}$$

For point cloud Y, the centered radial distances are scaled by the positive scale factor e^{ds}, i.e.,

$$g_s(t; ds) := \frac{1}{m} \sum_{k=1}^{m} \frac{1}{\sqrt{2\pi}\sigma_s} \exp\left[-\frac{\left(t - e^{ds} \left[\|y_k - \bar{y}\| - \|Y\|\right]\right)^2}{2\sigma_s^2} \right]. \tag{14}$$

Rotational KDE. To estimate rotation independently of translation, both point clouds are centered at their respective centroids. We construct three-dimensional isotropic Gaussian KDEs.

For point cloud X, we have

$$f_r(t) := \frac{1}{n} \sum_{k=1}^{n} \frac{1}{(2\pi)^{3/2}\sigma_r^3} \exp\left[-\frac{\|t - (x_k - \bar{x})\|^2}{2\sigma_r^2} \right]. \tag{15}$$

For point cloud Y, we have

$$g_r(t; dq) := \frac{1}{m} \sum_{k=1}^{m} \frac{1}{(2\pi)^{3/2}\sigma_r^3} \exp\left[-\frac{\|t - R_{dq}(y_k - \bar{y})\|^2}{2\sigma_r^2} \right], \tag{16}$$

D Expanded Loss Function

In this section, we derive the full loss function. Following the notation in the Appendix C, the total loss is given by

$$\mathcal{L} = \mathcal{L}_t(dt_1, dt_2, dt_3) + \mathcal{L}_s(ds) + \mathcal{L}_r(dq), \tag{17}$$

where $\mathcal{L}_t$, $\mathcal{L}_s$, and $\mathcal{L}_r$ are the loss functions for translation, scale, and rotation, respectively. Each is evaluated as the squared $\mathcal{L}^2$ distance between the corresponding KDEs, i.e.,

$$\mathcal{L}_t(dt_1, dt_2, dt_3) = \sum_{i=1}^{3} \int_{\mathbb{R}} |f_i(t) - g_i(t; dt_i)|^2 \, dt, \tag{18}$$

$$\mathcal{L}_s(ds) = \int_{\mathbb{R}} |f_s(t) - g_s(t; ds)|^2 \, dt, \tag{19}$$

$$\mathcal{L}_r(dq) = \int_{\mathbb{R}^3} |f_r(t) - g_r(t; dq)|^2 \, dt. \tag{20}$$

Using the formula for the integral of the product of two isotropic Gaussian functions with equal bandwidth σ (Rasmussen & Williams, 2005):

$$\int_{\mathbb{R}^d} \mathcal{N}(t; \mu_1, \sigma^2 I) \mathcal{N}(t; \mu_2, \sigma^2 I) \, dt = \frac{1}{(4\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\mu_1 - \mu_2\|^2}{4\sigma^2} \right), \tag{21}$$

where $\mathcal{N}(t; \mu, \sigma^2 I)$ denotes the probability density function of a d-dimensional isotropic Gaussian distribution with mean μ and covariance matrix $\sigma^2 I$, we can expand each loss function as follows.

For translation, the loss is

$$\mathcal{L}_t = \sum_{i=1}^{3} \left[\int_{\mathbb{R}} f_i(t)^2 dt + \int_{\mathbb{R}} g_i(t; dt_i)^2 dt - 2 \int_{\mathbb{R}} f_i(t) g_i(t; dt_i) dt \right] \tag{22}$$

$$\begin{aligned} &= \sum_{i=1}^{3} \left[\frac{1}{2\sqrt{\pi}\sigma_t n^2} \sum_{k=1}^{n} \sum_{j=1}^{n} \exp\left(-\frac{\left(\langle e_i, x_k \rangle - \langle e_i, x_j \rangle\right)^2}{4\sigma_t^2} \right) \right. \\ &\qquad + \frac{1}{2\sqrt{\pi}\sigma_t m^2} \sum_{k=1}^{m} \sum_{j=1}^{m} \exp\left(-\frac{\left(\langle e_i, y_k \rangle - \langle e_i, y_j \rangle\right)^2}{4\sigma_t^2} \right) \\ &\qquad \left. - \frac{1}{\sqrt{\pi}\sigma_t nm} \sum_{k=1}^{n} \sum_{j=1}^{m} \exp\left(-\frac{\left(\langle e_i, x_k \rangle - \langle e_i, y_j \rangle - dt_i\right)^2}{4\sigma_t^2} \right) \right]. \end{aligned} \tag{23}$$

For scale, the loss is

$$\mathcal{L}_s = \int_{\mathbb{R}} f_s(t)^2 dt + \int_{\mathbb{R}} g_s(t; ds)^2 dt - 2 \int_{\mathbb{R}} f_s(t) g_s(t; ds) dt \tag{24}$$

$$\begin{aligned} &= \frac{1}{2\sqrt{\pi}\sigma_s n^2} \sum_{k=1}^{n} \sum_{j=1}^{n} \exp\left[-\frac{\left(\|x_k - \bar{x}\| - \|x_j - \bar{x}\|\right)^2}{4\sigma_s^2} \right] \\ &\quad + \frac{1}{2\sqrt{\pi}\sigma_s m^2} \sum_{k=1}^{m} \sum_{j=1}^{m} \exp\left[-\frac{e^{2ds}\left(\|y_k - \bar{y}\| - \|y_j - \bar{y}\|\right)^2}{4\sigma_s^2} \right] \\ &\quad - \frac{1}{\sqrt{\pi}\sigma_s nm} \sum_{k=1}^{n} \sum_{j=1}^{m} \exp\left[-\frac{\left(\left[\|x_k - \bar{x}\| - \|X\|\right] - e^{ds}\left[\|y_j - \bar{y}\| - \|Y\|\right]\right)^2}{4\sigma_s^2} \right]. \end{aligned} \tag{25}$$

For rotation, the loss is

$$\mathcal{L}_r = \int_{\mathbb{R}^3} f_r(t)^2 dt + \int_{\mathbb{R}^3} g_r(t; dq)^2 dt - 2 \int_{\mathbb{R}^3} f_r(t) g_r(t; dq) dt \tag{26}$$

$$\begin{aligned} &= \frac{1}{8\pi^{3/2}\sigma_r^3 n^2} \sum_{k=1}^{n} \sum_{j=1}^{n} \exp\left[-\frac{\|x_k - x_j\|^2}{4\sigma_r^2} \right] \\ &\quad + \frac{1}{8\pi^{3/2}\sigma_r^3 m^2} \sum_{k=1}^{m} \sum_{j=1}^{m} \exp\left[-\frac{\|y_k - y_j\|^2}{4\sigma_r^2} \right] \\ &\quad - \frac{1}{4\pi^{3/2}\sigma_r^3 nm} \sum_{k=1}^{n} \sum_{j=1}^{m} \exp\left[-\frac{\|(x_k - \bar{x}) - R_{dq}(y_j - \bar{y})\|^2}{4\sigma_r^2} \right]. \end{aligned} \tag{27}$$

For translation, the first two terms are independent of dt, since a translation does not change pairwise distances within either point cloud. For scale, only the first term is independent of ds, since scaling changes the pairwise radial distances within Y. For rotation, the first two terms are independent of dq, since rotation preserves Euclidean distances.

Dropping terms that are unrelated to the learnable parameters dt, ds, and dq, we obtain

$$\mathcal{L}_t(dt_1, dt_2, dt_3) = -\sum_{i=1}^{3} \frac{1}{\sqrt{\pi}\sigma_t nm} \sum_{k=1}^{n}\sum_{j=1}^{m} \exp\left(-\frac{\left(\langle e_i, x_k\rangle - \langle e_i, y_j\rangle - dt_i\right)^2}{4\sigma_t^2}\right), \tag{28}$$

$$\begin{aligned}\mathcal{L}_s(ds) = &\frac{1}{2\sqrt{\pi}\sigma_s m^2} \sum_{k=1}^{m}\sum_{j=1}^{m} \exp\left[-\frac{e^{2ds}\left(\|y_k - \bar{y}\| - \|y_j - \bar{y}\|\right)^2}{4\sigma_s^2}\right] \\ &- \frac{1}{\sqrt{\pi}\sigma_s nm} \sum_{k=1}^{n}\sum_{j=1}^{m} \exp\left[-\frac{\left([\|x_k - \bar{x}\| - \|X\|] - e^{ds}\left[\|y_j - \bar{y}\| - \|Y\|\right]\right)^2}{4\sigma_s^2}\right],\end{aligned} \tag{29}$$

$$\mathcal{L}_r(dq) = -\frac{1}{4\pi^{3/2}\sigma_r^3 nm} \sum_{k=1}^{n}\sum_{j=1}^{m} \exp\left[-\frac{\left\|(x_k - \bar{x}) - R_{dq}(y_j - \bar{y})\right\|^2}{4\sigma_r^2}\right]. \tag{30}$$

The final loss function $\mathcal{L}$ is therefore the sum of Eqs. 28, 29, and 30.