Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Learning Point Cloud Geometry as a Statistical Manifold: Theory and Practice

Jinwoo Lee*[,1] Jiwoo Kim*[,1] Woojae Shin[1] Giseop Kim[2] Hyondong Oh[1,†]

[1] Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea

[2] Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Republic of Korea

Emails: {jinwoolee, tars0523, oj7987, h.oh}@kaist.ac.kr, gsk@dgist.ac.kr

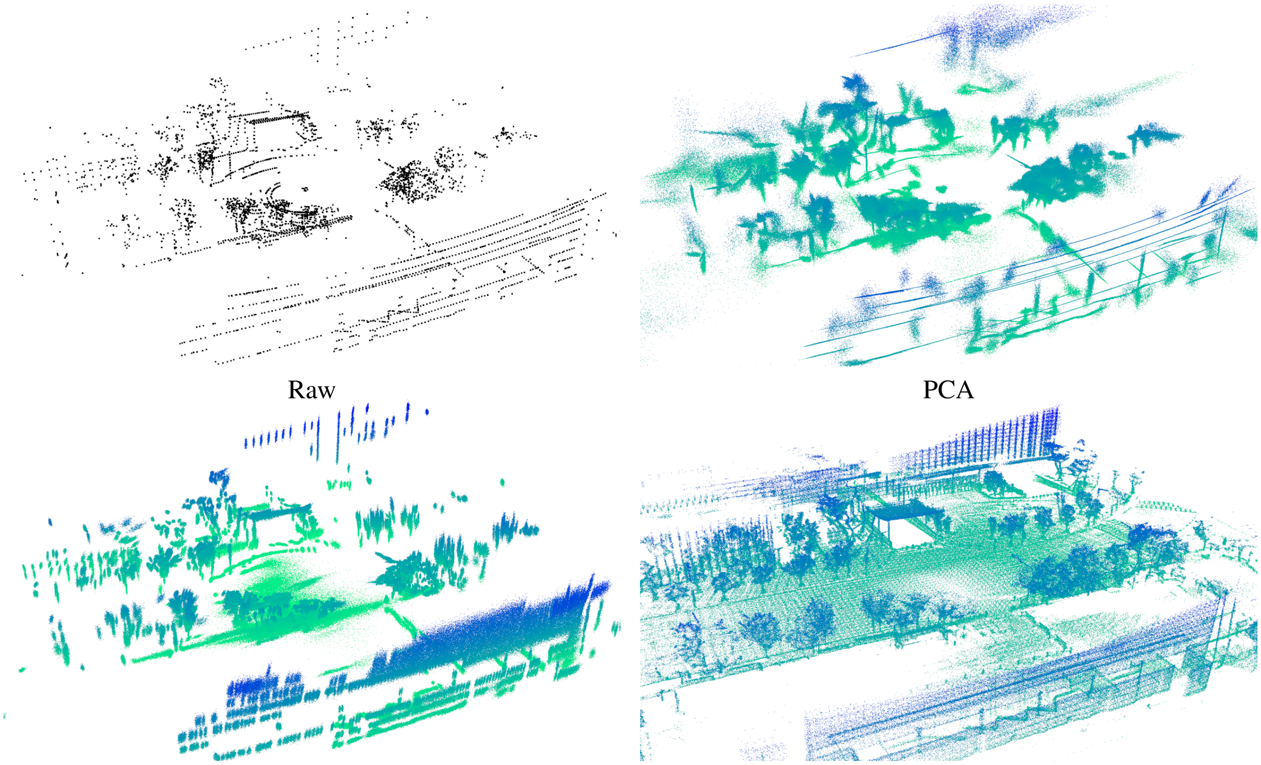


Fig. 1: Point cloud geometries estimated by different methods on the HeLiPR dataset, visualized with 300 samples per distribution. PCA struggles to recover underlying geometry, whereas POLI preserves geometry aligned with the ground truth.

Abstract*—Point clouds are a fundamental representation for robotic perception tasks such as localization, mapping, and object pose estimation. However, LiDAR-acquired point clouds are inherently sparse and non-uniform, providing incomplete observations of the underlying geometry. Such sparsity and non-uniformity hinder reliable geometric reasoning, leading to degraded performance in downstream perception tasks. To mitigate these issues, prior work has attempted to compensate for the sparsity and non-uniformity of point clouds by estimating point cloud geometry. However, in the absence of an explicit model of point cloud geometry, existing approaches have predominantly relied on either hand-crafted statistics of local point distributions or end-to-end supervised deep learning, which often suffer from limited scalability or require large amounts of accurately labeled training data. To address these challenges, we explicitly model and estimate point cloud geometry under a principled mathematical formulation. Theoretically, we represent the point cloud geometry as a statistical manifold induced by a family of Gaussian distributions that captures the local geometry of each point. Building on this formulation, we design a probabilistic model that predicts per-point local geometry in the form of a Gaussian distribution. Practically, we introduce a deep neural network to instantiate the estimation of these Gaussian distributions, and term the resulting estimator as Point-to-Ellipsoid (POLI). By consistently estimating point-wise local geometry across diverse point clouds, POLI learns a mapping between point cloud observations and their underlying geometry. Importantly, this mapping is learned in a self-supervised manner, removing the reliance on labeled data while maintaining strong geometric inductive biases. The resulting representation integrates seamlessly into existing robotic perception pipelines without requiring architectural modifications. Extensive experiments demonstrate that the proposed theory and practice enable accurate and robust estimation of point cloud geometry and consistently improve performance across a wide range of robotic perception tasks. The implementation of our proposed method is publicly available at https://github.com/jinwoolee1230/POLI.

*Equal contribution. †Corresponding author.

I. Introduction

Point clouds are widely used in robotic perception tasks such as localization, mapping, and object pose estimation [4, 22, 23, 30, 32, 35, 37, 46]. However, point clouds acquired by LiDAR are sparse and non-uniform, providing incomplete observations of the underlying geometry. This limits reliable geometric reasoning and degrades the performance of downstream perception tasks [17, 28, 36, 41].

To mitigate these limitations, existing robotic perception systems have fundamentally relied on estimating the underlying geometry from point clouds to enrich the available geometric information [30, 32]. Without an explicit model of point cloud geometry, most systems treat local point distributions as implicit representations of the underlying surface geometry and estimate point-wise local geometry using hand-crafted statistics over local neighborhoods [13, 23, 30, 32, 33, 35, 39, 40]. However, in practice, point distributions are strongly influenced by sensor resolution and viewpoint, reflecting observation artifacts rather than underlying geometry.

To overcome such heuristic limitations, end-to-end supervised deep learning approaches have been proposed [2, 3, 36, 41]. While these methods have shown improved scalability, they delegate most of the geometric reasoning to neural networks, resulting in strong dependence on large-scale annotated data and limited incorporation of geometric inductive biases. Moreover, their representations have limitations in being integrated into robotic downstream tasks, and it is difficult to guarantee consistent geometric representations [41].

In both cases, the lack of a principled model limits the accuracy and scalability of point cloud geometry estimation. By contrast, if the relationship between point clouds and underlying geometry is formulated in a principled mathematical way, accurate and scalable geometry estimation becomes possible, with the potential to benefit a wide range of robotic perception systems. Motivated by this perspective, we propose a principled mathematical formulation for point cloud geometry estimation, together with its theoretical foundations and a practical implementation.

Specifically, we model point cloud geometry as a statistical manifold induced by a family of three-dimensional (3D) Gaussian distributions, where each distribution encodes the local geometry associated with a point. Based on this formulation, we construct a principled probabilistic framework for estimating point-wise covariances and introduce deep learning as a practical instantiation of this framework. This learning framework equips neural networks with strong geometric inductive biases while enabling self-supervised learning without requiring accurate labels. The resulting covariance estimator, termed Point-to-Ellipsoid (POLI), learns a mapping between point cloud observations and their underlying geometry by consistently estimating point-wise local geometry across diverse point clouds. Moreover, POLI can be seamlessly integrated into existing robotic perception pipelines [16, 23, 40, 42] without requiring structural modifications.

II. Point Cloud Geometry Estimation

Point cloud geometry plays a fundamental role in robotic perception pipelines, as it enables informative representations of point clouds. Existing methods for point cloud geometry estimation can be broadly categorized into two classes: hand-crafted neighborhood statistics and deep learning–based approaches. In this section, we review representative methods from both categories and discuss their limitations.

A. Hand-Crafted Neighborhood Statistics

Hand-crafted neighborhood statistics, methods that approximate the underlying surface geometry using local point distributions within a neighborhood, were originally introduced to improve robotic perception performance. Notably, iterative closest point (ICP)-based registration methods estimate local geometry to construct reliable error metrics. Point-to-plane ICP [29] estimates a local planar surface from neighboring points, enabling a plane-based residual. GICP [32] represents local geometry using a covariance matrix obtained from the local point distribution, enabling an adaptive residual metric. VGICP [15] aggregates point-wise local statistics within voxel grids, enabling robust local geometry estimation.

In correspondence estimation, local geometry is estimated to obtain distinctive point features. PFH [30] and FPFH [31] encode local geometry using angular relationships between point normals and inter-point directions within a local neighborhood. SHOT [35] encodes local geometry by accumulating normal orientation distributions within a local reference frame, and RoPS [10] represents local geometry using statistics of local point distributions obtained from rotational projections.

While these statistical approaches have been effective across a wide range of applications, they estimate local geometry directly from observed point neighborhoods using statistical techniques such as principal component analysis (PCA). Consequently, their accuracy depends on the observation process, including sensor resolution and viewpoint. In addition, their performance is sensitive to neighborhood selection, limiting their scalability, as shown in Fig. 1.

B. Learning-Based Approaches

To improve the scalability of conventional methods, learning-based approaches have been proposed. Many existing works focus on point-wise surface normal estimation [19, 20, 21], which provides a compact description of local geometry. However, normals alone fail to capture geometric properties such as local planar extent or curvature, limiting their utility in downstream tasks.

Another line of work aims to densify point clouds or reconstruct complete surface geometry using generative models [17, 25, 28, 36, 41, 45]. While these methods can produce visually dense reconstructions, predicting geometry in unobserved regions often leads to inaccurate and inconsistent estimates. Moreover, they typically require dense scans for training, which are difficult to obtain in practice, and can also incur high computational cost. More discussion is provided in Supplementary Material SF.

Despite their empirical success, many learning-based approaches rely on fully end-to-end training without explicitly modeling the underlying geometric structure. As a result, geometric reasoning is implicitly delegated to neural networks, leading to a strong dependence on large amounts of carefully annotated training data and fundamental limitations in accuracy and scalability. From this perspective, we introduce a new problem formulation that models point cloud geometry as a statistical manifold, together with a solid theoretical foundation and a practical instantiation via neural networks. Crucially, our formulation enables self-supervised learning, requiring neither dense scans nor perfect transformation labels, thereby substantially expanding the range of datasets that can be leveraged for geometry estimation.

III. Theory: Modeling Point Cloud Geometry

In this section, we present a theoretical formulation that models point cloud geometry as a statistical manifold and derives a probabilistic framework for its estimation.

A. Point Cloud Geometry as a Statistical Manifold

A statistical manifold is a smooth manifold formed by a family of probability distributions parameterized by a continuous parameter space [1]. Since each element on a statistical manifold represents a probability distribution rather than a single point measurement, modeling local surface geometry with individual distributions provides an expressive representation of the underlying surface while reducing the reliance on dense observations. Building on this idea, we model point cloud geometry as a statistical manifold, denoted by $\mathcal{M}_{\mathbf{g}}$, induced by Gaussian distributions that encode local surface geometry. Beyond modeling, we formulate the corresponding estimation problem and provide a practical instantiation by adopting Gaussian distributions, which are analytically tractable and well-suited for compact geometric representation.

Definition 1 (Statistical Manifold of World Geometry)**.** *The statistical manifold $\mathcal{M}_{\mathbf{g}}$ represents the underlying world surface manifold through a probabilistic approximation in a global reference frame. The manifold $\mathcal{M}_{\mathbf{g}}$ is defined as a family of 3D Gaussian distributions, each parameterized by a mean $\bar{\mathbf{x}}_{\mathbf{g}_i} \in \mathbb{R}^3$ and a covariance matrix $\mathbf{C}_{\mathbf{g}_i} \in \mathbb{S}^3_+$. Each Gaussian distribution defines the probability density that an observed point $\mathbf{x}_i$ is measured with respect to a local tangent plane at $\bar{\mathbf{x}}_{\mathbf{g}_i}$.*

When $\mathbf{x}_i$ is acquired from a high-precision LiDAR sensor, the resulting Gaussian distribution approximates the local planar geometry as shown in Fig. 2. In this formulation, the covariance matrix $\mathbf{C}_{\mathbf{g}_i}$ provides a quantitative representation of the geometry [26] in the neighborhood of $\bar{\mathbf{x}}_{\mathbf{g}_i}$.

Proposition 1 (Statistical Manifold Representation from Different Sensor Frames)**.** *Given the sensor pose $\mathbf{T}_{\mathbf{p}} = (\mathbf{R}_{\mathbf{p}}, \mathbf{t}_{\mathbf{p}}) \in SE(3)$ relative to the global reference frame, the parameters $\bar{\mathbf{x}}_{\mathbf{p}_i}$ and $\mathbf{C}_{\mathbf{p}_i}$ of the local statistical manifold $\mathcal{M}_{\mathbf{p}}$ are defined by transforming the global model $\mathcal{M}_{\mathbf{g}}$ into the sensor frame via ${\mathbf{T}_{\mathbf{p}}}^{-1}$:*

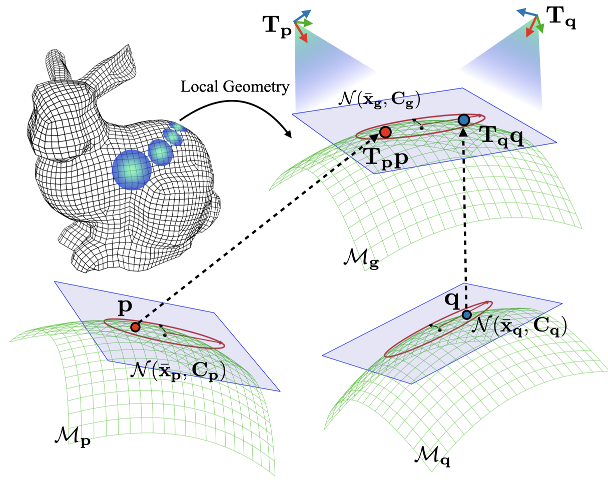


Fig. 2: A visualization illustrating how underlying geometry can be represented as a statistical manifold. It explains that two points observed in different coordinate frames can be regarded as samples drawn from the same underlying distribution.

$$\bar{\mathbf{x}}_{\mathbf{p}_i} \triangleq \mathbf{R}_{\mathbf{p}}^{\top}(\bar{\mathbf{x}}_{\mathbf{g}_i} - \mathbf{t}_{\mathbf{p}}), \qquad \mathbf{C}_{\mathbf{p}_i} \triangleq \mathbf{R}_{\mathbf{p}}^{\top}\mathbf{C}_{\mathbf{g}_i}\mathbf{R}_{\mathbf{p}}.$$

Analogously, for a second observation frame with pose $\mathbf{T}_{\mathbf{q}} = (\mathbf{R}_{\mathbf{q}}, \mathbf{t}_{\mathbf{q}}) \in SE(3)$, the parameters $\bar{\mathbf{x}}_{\mathbf{q}_i}$ and $\mathbf{C}_{\mathbf{q}_i}$ of the statistical manifold $\mathcal{M}_{\mathbf{q}}$ are defined via ${\mathbf{T}_{\mathbf{q}}}^{-1}$:

$$\bar{\mathbf{x}}_{\mathbf{q}_i} \triangleq {\mathbf{R}_{\mathbf{q}}}^{\top}(\bar{\mathbf{x}}_{\mathbf{g}_i} - \mathbf{t}_{\mathbf{q}}), \qquad \mathbf{C}_{\mathbf{q}_i} \triangleq {\mathbf{R}_{\mathbf{q}}}^{\top}\mathbf{C}_{\mathbf{g}_i}\mathbf{R}_{\mathbf{q}}.$$

Let $\mathcal{P} \triangleq (\mathbf{p}_1, \ldots, \mathbf{p}_N)$ and $\mathcal{Q} \triangleq (\mathbf{q}_1, \ldots, \mathbf{q}_N)$ denote two scans acquired from frames $\mathbf{T}_{\mathbf{p}}$ and $\mathbf{T}_{\mathbf{q}}$, respectively, with known correspondences $(\mathbf{p}_i, \mathbf{q}_i)$. Each point is independently sampled as a noisy observation from its local Gaussian distribution, i.e., $\mathbf{p}_i \sim \mathcal{N}(\bar{\mathbf{x}}_{\mathbf{p}_i}, \mathbf{C}_{\mathbf{p}_i})$ and $\mathbf{q}_i \sim \mathcal{N}(\bar{\mathbf{x}}_{\mathbf{q}_i}, \mathbf{C}_{\mathbf{q}_i})$, with per-point covariances $\mathbf{C}_{\mathbf{p}} \triangleq (\mathbf{C}_{\mathbf{p}_1}, \ldots, \mathbf{C}_{\mathbf{p}_N})$ and $\mathbf{C}_{\mathbf{q}} \triangleq (\mathbf{C}_{\mathbf{q}_1}, \ldots, \mathbf{C}_{\mathbf{q}_N})$. Note that, $\mathcal{P}$ and $\mathcal{Q}$ originate from manifolds related by an isometric rigid-body transformation of the same underlying world surface, they share identical local geometry (i.e., $\mathbf{C}_{\mathbf{g}_i}$). Under this local geometric model, the displacement $\mathbf{d}_i$ for the i-th pair is defined by the relative transformation $\mathbf{T} \triangleq \mathbf{T}_{\mathbf{q}}^{-1}\mathbf{T}_{\mathbf{p}}$ as:

$$\mathbf{d}_i \triangleq \mathbf{q}_i - \mathbf{T}\mathbf{p}_i, \qquad \mathbf{d}_i \sim \mathcal{N}(\mathbf{0},\, 2\mathbf{C}_{\mathbf{q}_i})\,. \tag{1}$$

Derivation of Eq. (1) is provided in Supplementary Material SA.

Lemma 1 (Re-parameterization)**.** *Given the relative transformation $\mathbf{T}$, the conditional likelihood of a target point $\mathbf{q}_i$ given its source point $\mathbf{p}_i$ depends only on the residual in Eq.* (1)*:*

$$p(\mathbf{q}_i \mid \mathbf{p}_i, \mathbf{T}, \mathbf{C}_{\mathbf{q}_i}) \equiv p(\mathbf{d}_i \mid \mathbf{T}, \mathbf{C}_{\mathbf{q}_i}). \tag{2}$$

This re-parameterization is justified by an invertible change of variables with unit Jacobian. Under the local geometric model in Eq. (1)*, we obtain:*

$$p(\mathbf{q}_i \,|\, \mathbf{p}_i, \mathbf{T}, \mathbf{C}_{\mathbf{q}_i}) = \frac{1}{\sqrt{(2\pi)^3\,|2\mathbf{C}_{\mathbf{q}_i}|}} \exp\!\left(-\tfrac{1}{2}\,\mathbf{d}_i^{\top}(2\mathbf{C}_{\mathbf{q}_i})^{-1}\mathbf{d}_i\right).$$

Derivation of Eq. (2) is provided in Supplementary Material SA.

Proposition 2 (Factorized Correspondence Likelihood). *The conditional likelihood of the target set $\mathcal{Q}$ given the fixed source set $\mathcal{P}$ factorizes as:*

$$\begin{aligned} p(\mathcal{Q} \,|\mathcal{P}, \mathbf{T}, \mathbf{C_q}) &= \prod_{i=1}^{N} p(\mathbf{q}_i \mid \mathbf{p}_i, \mathbf{T}, \mathbf{C}_{\mathbf{q}_i}) \\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{(2\pi)^3\,|2\mathbf{C}_{\mathbf{q}_i}|}} \exp\Big(-\frac{1}{2}\,\mathbf{d}_i^\top (2\mathbf{C}_{\mathbf{q}_i})^{-1}\mathbf{d}_i\Big), \end{aligned} \tag{3}$$

where the last equality follows from Lemma 1.

B. Probabilistic Modeling for Covariance Estimation

Under a probabilistic model that relates two scans acquired from the same underlying scene, their relative pose, and the local geometry, we derive a maximum likelihood estimation (MLE) framework. Within this framework, we formulate an optimization problem to estimate the underlying local geometry in terms of point-wise covariances.

Assumption 1 (Probabilistic Modeling). *We aim to estimate the covariances $\mathbf{C_q}$ from observations $\mathcal{D} = (\mathcal{Q}, \tilde{\mathbf{T}})$, conditioned on the source set $\mathcal{P}$. Here, $\tilde{\mathbf{T}}$ denotes a relative pose measurement between the sensor frames at which the scans are acquired, representing an observation of the underlying transformation $\mathbf{T}$ (e.g., a relative pose measured by a GNSS–INS system). We write the marginal likelihood as:*

$$\begin{aligned} p(\mathcal{D}\,|\,\mathcal{P}, \mathbf{C_q}) &= \int p(\mathcal{D}, \mathbf{T} \mid \mathcal{P}, \mathbf{C_q})\, d\mathbf{T} \\ &\qquad \textit{(marginalization)} \\ &= \int p(\mathcal{Q} \,|\tilde{\mathbf{T}}, \mathcal{P}, \mathbf{T}, \mathbf{C_q})\; p(\tilde{\mathbf{T}} \mid \mathbf{T}, \mathcal{P}, \mathbf{C_q})\; p(\mathbf{T} \mid \mathcal{P}, \mathbf{C_q})\, d\mathbf{T} \\ &\qquad \textit{(chain rule)} \\ &= \int p(\mathcal{Q} \,|\mathcal{P}, \mathbf{T}, \mathbf{C_q})\; p(\tilde{\mathbf{T}} \mid \mathbf{T})\; p(\mathbf{T})\, d\mathbf{T} \\ &\qquad (\textit{conditional independence}) \\ &\propto \int p(\mathcal{Q} \,|\mathcal{P}, \mathbf{T}, \mathbf{C_q}) p(\tilde{\mathbf{T}} \mid \mathbf{T})\, d\mathbf{T}. \\ &\qquad \textit{(uniform prior for } p(\mathbf{T})\textit{)} \end{aligned} \tag{4}$$

We model the relationships among the variables as follows. The latent pose $\mathbf{T}$ is assumed to be independent of $(\mathcal{P}, \mathbf{C_q})$, and we adopt a non-informative prior $p(\mathbf{T}) \propto 1$. The observed pose $\tilde{\mathbf{T}}$ depends solely on the latent pose $\mathbf{T}$, and is conditionally independent of $\mathcal{Q}$ given $(\mathcal{P}, \mathbf{T}, \mathbf{C_q})$. Intuitively, once $\mathcal{P}$ is given, the latent pose $\mathbf{T}$—representing the relative transformation between $\mathcal{P}$ and $\mathcal{Q}$—serves as the primary factor driving both $\tilde{\mathbf{T}}$ and $\mathcal{Q}$. Specifically, $\tilde{\mathbf{T}}$ is generated directly from a realization of $\mathbf{T}$, while $\mathcal{Q}$ is determined by the combination of $\mathbf{T}$ and the intrinsic factor $\mathbf{C_q}$. These dependency relationships are illustrated in Fig. 3. We assume that any potential dependencies beyond those explicitly modeled are negligible, allowing us to focus on the key dependencies in our formulation. Further discussion is provided in Supplementary Material SE.

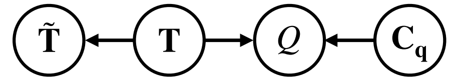


Fig. 3: Probabilistic model as a directed acyclic graph, where edges denote direct conditional dependencies.

Definition 2 (Energy Function). *To simplify the expression of $p(\mathcal{D} \mid \mathcal{P}, \mathbf{C_q})$ and facilitate the derivation of its MLE, we derive the energy function of the integrand in Eq. 4, i.e., the negative log-probability of the corresponding distribution. For the pose likelihood (i.e., $p(\tilde{\mathbf{T}} \mid \mathbf{T})$), we assume a Gaussian noise model:*

$$p(\tilde{\mathbf{T}} \mid \mathbf{T}) = \frac{1}{(2\pi)^3\,|\Gamma|^{1/2}} \exp\left(-\frac{1}{2}\,\xi(\mathbf{T})^\top \Gamma^{-1}\,\xi(\mathbf{T})\right), \tag{5}$$

where $\xi(\mathbf{T}) = \operatorname{Log}\Big(\mathbf{T}^{-1}\tilde{\mathbf{T}}\Big) \in \mathbb{R}^6$ and Γ denotes the covariance of the measurement noise. Combining Eqs. (3) and (5), we define the energy:

$$\begin{aligned} \Phi(\mathbf{T}; \mathbf{C_q}) &\triangleq -\log\Big(p(\mathcal{Q} \mid \mathcal{P}, \mathbf{T}, \mathbf{C_q})\; p(\tilde{\mathbf{T}} \mid \mathbf{T})\Big) \\ &= \frac{1}{2}\sum_{i=1}^{N}\Big[\log|2\mathbf{C}_{\mathbf{q}_i}| + \mathbf{d}_i^\top(2\mathbf{C}_{\mathbf{q}_i})^{-1}\mathbf{d}_i\Big] \\ &\quad + \frac{1}{2}\log|\Gamma| + \frac{1}{2}\,\xi(\mathbf{T})^\top\Gamma^{-1}\xi(\mathbf{T}) + \textit{const.} \end{aligned} \tag{6}$$

Then, the marginal likelihood from Assumption 1 can be written as:

$$\begin{aligned} p(\mathcal{D} \mid \mathcal{P}, \mathbf{C_q}) &\propto \int p(\mathcal{Q} \mid \mathcal{P}, \mathbf{T}, \mathbf{C_q})\; p(\tilde{\mathbf{T}} \mid \mathbf{T})\, d\mathbf{T} \\ &\propto \int \exp\big(-\Phi(\mathbf{T}; \mathbf{C_q})\big)\, d\mathbf{T}. \end{aligned} \tag{7}$$

Proposition 3 (Laplace Approximation). *The integral in Eq. (7) is generally intractable due to the nonlinearity of $\Phi(\mathbf{T}; \mathbf{C_q})$ over $SE(3)$. We therefore adopt a Laplace approximation, which approximates the integral by fitting a Gaussian to the integrand around its mode (most probable pose $\hat{\mathbf{T}}$). This yields a closed-form, differentiable surrogate objective with stable gradients, avoiding the computational overhead of sampling-based alternatives [24]. Since the integrand is proportional to $\exp\big(-\Phi(\mathbf{T}; \mathbf{C_q})\big)$, the dominant contribution comes from the neighborhood of the minimizer of Φ. Accordingly, we first compute the mode pose*

$$\hat{\mathbf{T}} \triangleq \arg\min_{\mathbf{T}} \Phi(\mathbf{T}; \mathbf{C_q}), \tag{8}$$

and apply a second-order Taylor expansion of Φ around $\hat{\mathbf{T}}$

$$p(\mathcal{D} \mid \mathcal{P}, \mathbf{C_q}) \approx K_0\,(2\pi)^3\,\big|H(\hat{\mathbf{T}})\big|^{-1/2} \exp\big(-\Phi(\hat{\mathbf{T}}; \mathbf{C_q})\big), \tag{9}$$

where K_0 is constant and $H(\hat{\mathbf{T}}) \in \mathbb{R}^{6\times 6}$ denotes the Hessian of Φ with respect to a $SE(3)$ perturbation, evaluated at $\hat{\mathbf{T}}$:

$$H(\hat{\mathbf{T}}) \triangleq \nabla^2_{\mathbf{T}} \Phi(\mathbf{T}; \mathbf{C_q})\big|_{\mathbf{T}=\hat{\mathbf{T}}}. \tag{10}$$

Corollary 1 (Geometry Estimation)**.** *Consequently, the MLE for the covariances is obtained by maximizing the approximate marginal likelihood, equivalently*

$$\hat{\mathbf{C}}_q = \arg\max_{\mathbf{C_q}} \; p(\mathcal{D} \mid \mathcal{P}, \mathbf{C_q}) \tag{11}$$

$$\approx \arg\min_{\mathbf{C_q}} \frac{1}{2} \log |H(\hat{\mathbf{T}})| + \Phi(\hat{\mathbf{T}}; \mathbf{C_q}). \tag{12}$$

With this probabilistic formulation, we estimate point cloud geometry in the form of point-wise covariances $\mathbf{C_q}$, which are elements of the statistical manifold representing the underlying geometry. This probabilistic framework only requires reasonably accurate relative pose measurements $\tilde{\mathbf{T}}$ and consistent correspondences $(\mathbf{p}_i, \mathbf{q}_i)$, enabling self-supervised learning without relying on dense point clouds or explicit ground-truth geometry in the following section.

IV. Practice: Learning Point Cloud Geometry

In this section, we describe how to solve the optimization problem in Corollary 1. The covariance set to be optimized, $\mathbf{C_q} \in \mathbb{R}^{6\times N}$, which comprises six parameters per covariance across N correspondences, requires a high-dimensional, nonconvex optimization problem that is impractical to solve using classical optimization methods. Therefore, we reformulate the problem as learning a mapping from the input point cloud $\mathcal{Q}$ to the set of covariance matrix $\mathbf{C_q}$. Specifically, we train a neural network to predict $\mathbf{C_q}$ such that the objective in Eq. (12) is minimized. The remainder of this section is organized as follows: We first present a neural network architecture of a covariance estimator that predicts point-wise covariances, then introduce a certifiably correct and differentiable optimization layer for the mode pose estimation and gradient calculation, and finally summarize the overall training and inference procedure. Through this formulation, we enable the network to infer the underlying geometry of the point cloud input as a set of point-wise covariances that constitute the statistical manifold.

A. Covariance Estimator

A neural network that learns to predict point-wise covariance must be permutation invariant, extract local features in the underlying geometry, and handle variable-sized and nonuniform point sets. Therefore, we design a neural network inspired by PointNet++ [27] to predict a per-point covariance matrix. Given a point cloud, a subset of points is selected via farthest point sampling, and neighborhoods are queried around them using a ball query. Each neighborhood is normalized to its centroid frame and encoded by applying a shared multi layer perceptron (MLP) to each point individually, followed by max pooling to obtain a fixed-length representation of the region. The resulting local features are back-propagated to the original point cloud using distance-weighted interpolation and skip connections, producing point-wise feature representations. For each point $\mathbf{q}_i$, the network predicts six parameters that define a lower-triangular matrix $\mathbf{L}_i$. The covariance is then constructed as $\mathbf{C}_{\mathbf{q}_i} = \mathbf{L}_i \mathbf{L}_i^\top$, which guarantees $\mathbf{C}_{\mathbf{q}_i} \succeq 0$. Network architecture is illustrated in Fig. 4

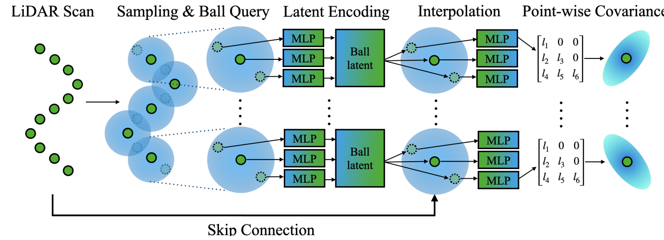


Fig. 4: Network architecture of covariance estimator.

B. Certifiably Correct Differentiable Optimization Layer

Given network-predicted covariances $\mathbf{C_q}$, we should estimate the mode pose in Eq. (8):

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T}} \sum_{i=1}^{N} \mathbf{d}_i^\top \left(2\mathbf{C}_{\mathbf{q}_i}\right)^{-1} \mathbf{d}_i + \xi(\mathbf{T})^\top \Gamma^{-1} \xi(\mathbf{T}). \tag{13}$$

However, to the best of our knowledge, solving Eq. (13) is challenging due to the nonlinearity introduced by $SE(3)$ and the anisotropy of the weighting factor [11, 12]. Moreover, even when using iterative solvers, convergence to the true optimum is not guaranteed, so gradients obtained by implicit function theorem (IFT) [5] may be biased toward suboptimal solutions [12]. Accordingly, we relax Eq. (13) to:

$$\hat{\mathbf{T}} \approx \arg\min_{\mathbf{T}} \sum_{i=1}^{N} \mathbf{d}_i^\top \left(2\mathbf{C}_{\mathbf{q}_i}\right)^{-1} \mathbf{d}_i. \tag{14}$$

This relaxation is well suited to our training setup, where the correspondence set $\mathcal{C} \triangleq (\mathcal{P}, \mathcal{Q})$ is constructed under a reference pose $\tilde{\mathbf{T}}$ by selecting nearest-neighbor pairs between $\mathcal{P}$ and $\mathcal{Q}$ and rejecting pairs whose distances exceed a predefined threshold. As a result, information about $\tilde{\mathbf{T}}$ is implicitly embedded in $\mathcal{C}$ and continues to guide the optimization even after the relaxation. Although this relaxation does not exactly solve the original problem, it enables us to leverage SDPRLayer [12].

SDPRLayer [12] provides certifiably globally optimal solutions, supports differentiation through the optimizer (allowing gradients to be computed at the optimum), and is computationally efficient. By making the inner pose estimation step both certifiably optimal and differentiable, we can train (i.e., backpropagate) the entire network end-to-end on large-scale datasets in a self-supervised manner. Details on the certifiable global optimization and differentiation are provided in Supplementary Material SB, while the relaxation leading to Eq. (14) is described in Supplementary Material SC.

C. Overall Training and Inference Procedure

Algorithms 1 and 2 summarize the training and inference procedures of POLI, respectively. During training, the neural network predicts a point-wise lower-triangular matrix $\mathbf{L}$ from the target point cloud $\mathcal{Q}$, and the covariance set is constructed as $\mathbf{C_q} = \mathbf{L}\mathbf{L}^\top$. Given $\mathbf{C_q}$ and the correspondence set $\mathcal{C}$, SDPRLayer [12] estimates the relative transformation $\hat{\mathbf{T}}$ between $\mathcal{P}$ and $\mathcal{Q}$ in a differentiable manner. The network is then optimized by back-propagating the proposed loss through

Algorithm 1 Training Stage of POLI (**PO**int-to-el**LI**psoid)

Require: Neural Network $\texttt{NN}(\cdot;\mathbf{\Theta})$, $\mathcal{C}$, $\tilde{\mathbf{T}}$, Learning Rate η, point cloud $\mathcal{P}$ and point cloud $\mathcal{Q}$

Ensure: Updated parameters $\mathbf{\Theta}$

/* Predict Lower-Triangular Matrices */

$\mathbf{L} \leftarrow \texttt{NN}(\mathcal{Q};\mathbf{\Theta})$

/* Construct Covariances */

$\mathbf{C_q} \leftarrow \mathbf{L}\mathbf{L}^\top$

/* Certifiably Correct Differentiable Optimization */

$\hat{\mathbf{T}}, \frac{\mathbf{d}\hat{\mathbf{T}}}{\mathbf{dC_q}} \leftarrow \texttt{SDPRLayer}(\mathbf{C_q}, \mathcal{C}, \mathcal{P}, \mathcal{Q})$

/* Loss Calculation */

$\mathcal{L}(\mathbf{\Theta}) = \Phi(\hat{\mathbf{T}}; \mathbf{C}_q) + \frac{1}{2}\log|H(\hat{\mathbf{T}})|$

/* Neural Network Update */

$\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \eta\nabla_{\mathbf{\Theta}}\mathcal{L}$

Algorithm 2 Inference Stage of POLI

Require: Trained Neural Network $\texttt{NN}(\cdot;\mathbf{\Theta}^*)$, point cloud $\mathcal{Q}$

Ensure: Estimated covariance $\mathbf{C_q}$

/* Predict Lower-Triangular Matrices */

$\mathbf{L} \leftarrow \texttt{NN}(\mathcal{Q};\mathbf{\Theta}^*)$

/* Construct Covariances */

$\mathbf{C_q} \leftarrow \mathbf{L}\mathbf{L}^\top$

return $\mathbf{C_q}$

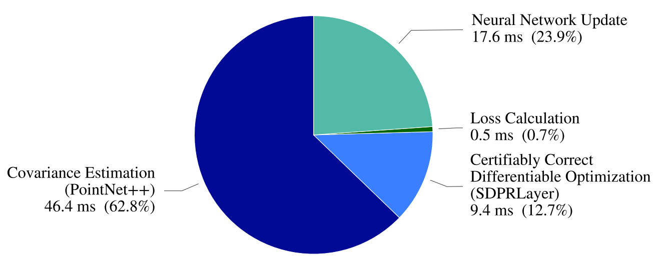


Fig. 5: Runtime composition of POLI training pipeline.

this differentiable optimization layer. The total training time is reported in Fig. 5. For an input containing approximately 16,000 points, each training step requires 73.9 ms.

During inference, the trained network predicts lower-triangular matrices $\mathbf{L}$ for $\mathcal{Q}$, from which the point-wise covariance sets are obtained as $\mathbf{C_q} = \mathbf{L}\mathbf{L}^\top$. The resulting covariances serve as local geometry for downstream tasks.

V. Experiments

POLI learns to predict the underlying geometry of point clouds by estimating point-wise covariance, making it a versatile module for a broad range of robotic perception tasks. In this section, we evaluate the effectiveness of POLI when integrated into robotic perception tasks: object pose estimation, LiDAR odometry, and scan augmentation. By leveraging accurately estimated point-wise covariance, we integrate it into existing pipelines through multiple representations.

For object pose estimation, we incorporate the point-wise normals predicted by POLI into an FPFH-based local descriptor [30] and evaluate its effectiveness by measuring the registration success rate. For LiDAR scan matching, we integrate the covariance predicted by POLI into the GICP [32] framework and evaluate its LiDAR odometry performance. We also assess POLI's scan augmentation capability by measuring downstream performance improvements when POLI-based augmentation is applied to robotic perception pipelines. Detailed experimental setups and results are provided in the Supplementary Material SD.

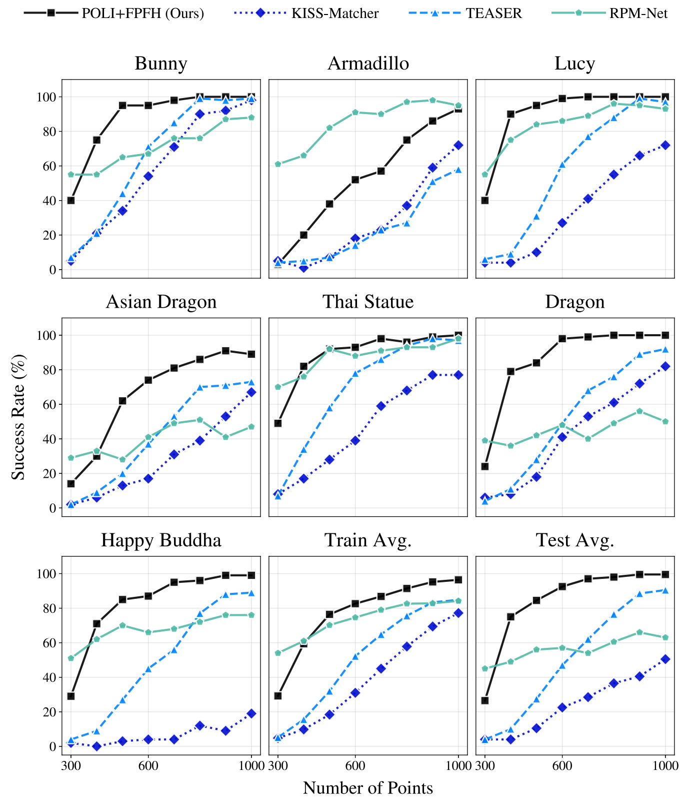


Fig. 6: Performance comparison on object pose estimation across different sparsity levels.

A. Object Pose Estimation

Setup. We evaluated POLI on the Stanford 3D Scanning Repository [6]. The training split consists of *Bunny, Armadillo, Asian Dragon, Lucy,* and *Thai Statue*, while the test split consists of *Dragon* and *Happy Buddha*. For training, we generated scan pairs by independently sampling $N \in \{500, 1{,}000\}$ points from the training objects and applying random axis–angle rotations up to 60°. For testing, we generated point clouds from the test objects with $N \in \{300, 400, 500, 600, 700, 800, 900, 1{,}000\}$ points, using 100 pairs per object. For correspondence estimation, we followed the Open3D FPFH pipeline [47], using POLI normals (the eigenvector of $\mathbf{C_q}$ along its smallest eigenvalue) and selecting inliers via ROBIN [34]. As baselines, we evaluated KISS-Matcher [23], TEASER [42], and RPM-Net [43] (Note that RPM-Net is trained on a large dataset and involves iterative registration).

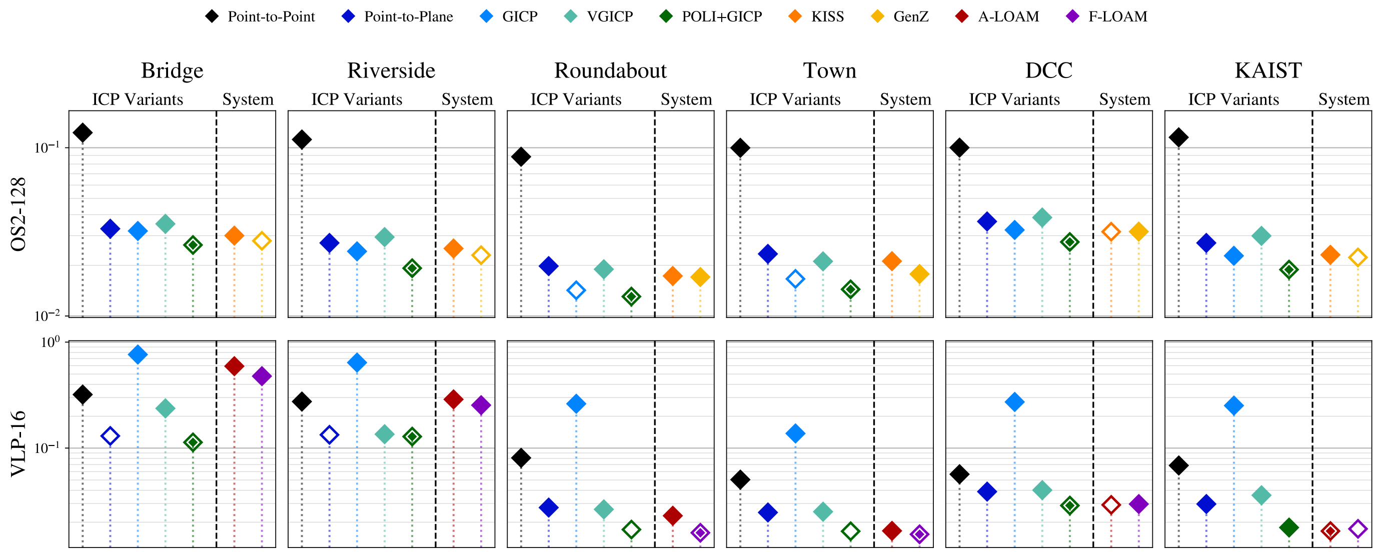


Fig. 7: Comparison of LiDAR odometry performance across different algorithms based on RPE (m). The best results are marked with diamonds enclosed by rings, and the second-best results are marked with hollow diamonds.

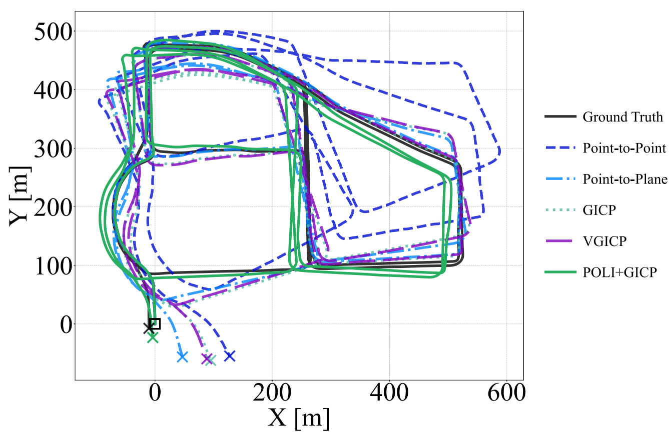


Fig. 8: Comparison of LiDAR odometry trajectories for different ICP variants on the DCC04 sequence.

Result. A registration is considered successful if the rotation error is $\leq 10°$ [22]. As shown in Fig. 6, POLI+FPFH achieves higher registration success rates across most settings and sparsity levels, and generalizes to the unseen *Dragon* and *Happy Buddha*, indicating benefits of accurate geometry estimation.

B. LiDAR Odometry

Setup. We conducted our LiDAR scan matching experiments on the HeLiPR dataset [14]. The *Roundabout*, *Town*, *Riverside*, and *Bridge* sequences were used for training, while the *KAIST* and *DCC* sequences were reserved for testing. Notably, the sequences in HeLiPR [14] exhibit no spatial overlap. Experiments were performed using two LiDAR sensors: the Velodyne VLP-16 and the Ouster OS2-128.

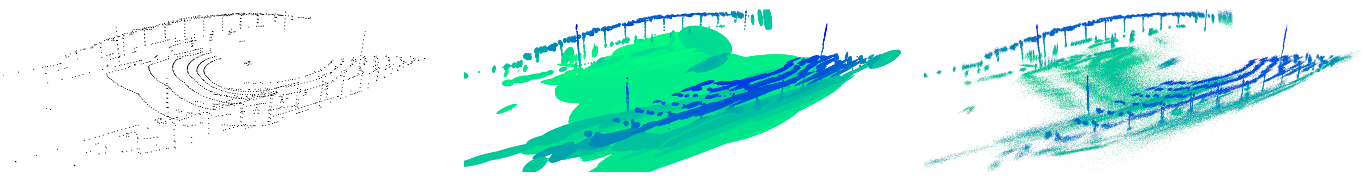
Voxel grid downsampling was applied with voxel sizes of 0.2 m for the VLP-16 and 1.0 m for the OS2-128. As baselines, we evaluated point-to-point ICP [4], point-to-plane ICP [29], GICP [32], and VGICP [15] under the same experimental settings. In addition, we compared our method with LiDAR odometry systems. For the VLP-16, A-LOAM [44] and F-LOAM [38] were used, while for the OS2-128, KISS-ICP [37] and GenZ-ICP [18] were evaluated. To assess the original performance of these LiDAR odometry systems, we used their default configurations.

Result. Performance was evaluated using relative pose error (RPE), as shown in Fig. 7. The proposed method consistently outperformed classical ICP-based approaches across all sequences. Furthermore, it achieved performance comparable to LiDAR odometry systems. Despite being evaluated on unseen environments (*DCC* and *KAIST*), POLI+GICP demonstrated consistent performance, showing its generalizability as shown in Fig. 8. These results indicate that estimating local geometry with POLI and using it for scan matching is highly beneficial.

C. Scan Augmentation

A key advantage of POLI is that it models point-cloud geometry probabilistically via estimated covariances, enabling principled scan augmentation through sampling and mathematically grounded densification (Fig. 9(a,b,c) and Fig. S.4). This allows us to densify LiDAR scans and plug it into existing robotic perception pipelines with minimal integration effort. We validate this via cross-dataset experiments on KITTI [8] and HeLiPR [14], focusing on LiDAR odometry and global registration. For both LiDAR odometry and scan registration, we use the VLP-16 scan dataset from HeLiPR and the 16-beam setting of KITTI used in [41]. For global registration, we randomly sample scan pairs within a 12 m range. For POLI, we employ the network trained in Section V-B on the HeLiPR [14] VLP-16 dataset.

The results are summarized in Tables I, and II, and visualized in Figs. S.2 and 3. Here, "+ Method" (e.g., + POLI) denotes applying the downstream pipeline to point clouds densi-



(a) Original point cloud (b) Estimated covariances from POLI (c) Augmented scan from POLI

Fig. 9: Scan augmentation process using POLI: raw scan, estimated covariances, and augmented scan generated by sampling.

TABLE I: LiDAR odometry results with augmentation methods, evaluated using RPE (m) at $\Delta = 100$ [9]. The best results are shown in bold, and the second-best results are underlined.

	HeLiPR [14]						KITTI [8]										
Method	*Bridge*	*Riverside*	*Roundabout*	*Town*	*DCC*	*KAIST*	00	01	02	03	04	05	06	07	08	09	10
KISS-ICP [37]	14.94	8.95	5.13	5.37	9.74	3.83	<u>2.50</u>	<u>14.82</u>	<u>2.98</u>	**1.77**	2.49	<u>2.19</u>	3.89	<u>1.50</u>	<u>3.44</u>	2.71	<u>2.03</u>
+PCA [47]	14.37	<u>7.82</u>	5.00	5.13	11.21	4.28	9.54	14.37	21.95	4.74	2.18	4.64	16.44	4.90	5.06	5.76	4.20
+TULIP [41]	<u>12.80</u>	15.48	<u>3.52</u>	<u>3.27</u>	<u>6.60</u>	<u>2.59</u>	2.76	**6.78**	16.79	2.36	**1.33**	2.57	**3.52**	1.83	3.75	<u>2.55</u>	2.76
+POLI	**3.89**	**1.92**	**1.16**	**1.03**	**1.77**	**0.95**	**2.32**	16.24	**2.95**	<u>1.78</u>	<u>2.01</u>	**1.97**	<u>3.58</u>	**1.30**	**3.42**	**2.39**	**1.66**

TABLE II: Success rate (%) of global registration using augmentation methods. The best results are shown in bold, and the second-best results are underlined.

HeLiPR [14]				
Method	Raw	+ PCA [47]	+ TULIP [41]	**+ POLI**
KISS-Matcher [23]	65.33%	<u>66.67%</u>	46.67%	**75.67%**
TEASER [42]	<u>74.33%</u>	72.33%	61.00%	**81.67%**
KITTI [8]				
Method	Raw	+ PCA [47]	+ TULIP [41]	**+ POLI**
KISS-Matcher [23]	54.55%	55.64%	**65.09%**	<u>60.73%</u>
TEASER [42]	64.36%	63.09%	<u>71.82%</u>	**79.09%**

TABLE III: Global registration success rate (%) across different resolutions on [8]. The best results are shown in bold, and the second-best results are underlined.

	KISS-Matcher [23]			TEASER [42]		
Resolution	Raw	**+ POLI**	Gain (%)	Base	**+ POLI**	Gain (%)
16 beams	54.55	<u>60.73</u>	+11.3	64.36	**79.09**	+22.9
32 beams	**87.27**	<u>86.18</u>	−1.3	<u>71.82</u>	**92.10**	+28.2
64 beams	<u>94.91</u>	**96.91**	+2.1	**97.82**	<u>97.64</u>	−0.2

TABLE IV: RPE and absolute pose error (APE) (m) [9] in *DCC* [14]. The best results are shown in bold.

	FAST-LIO [40]			GLIM [16]		
Metric ([m], [°])	Raw	**+ POLI**	Gain (%)	Raw	**+ POLI**	Gain (%)
RPE (Δ=100) Trans.	4.734	**3.200**	−32.4	7.006	**6.864**	−2.0
RPE (Δ=100) Rot.	5.296	**3.180**	−40.0	7.649	**7.390**	−3.4
APE Trans.	101.05	**44.79**	−55.7	56.94	**35.61**	−37.5
APE Rot.	36.61	**12.80**	−65.1	17.23	**10.06**	−41.6

fied using the corresponding augmentation method. Compared against PCA [47] and TULIP [41] (state-of-the-art LiDAR upsampling algorithm trained in KITTI [8]), POLI consistently performs best on both KITTI [8] and HeLiPR [14]. While TULIP [41] suffers from performance degradation under unseen data, POLI remains robust to variations in sensor type, field of view, and mounting configuration (e.g., the heavily occluded frontal region in HeLiPR [14]), highlighting the benefit of its strong geometric inductive bias induced by the combination of principled modeling and deep learning.

In addition, we evaluate global registration performance on the KITTI [8] dataset under varying LiDAR resolutions by downsampling the original scans to 16-, 32-, and 64-channel configurations. Importantly, no resolution-specific models are trained; instead, a single model is used across all LiDAR resolutions. As shown in Table III, POLI improves performance in most cases, demonstrating strong robustness to resolution changes. In particular, the performance gains are more pronounced in low-resolution settings, indicating that POLI effectively mitigates information loss caused by sparse sensing. Note that POLI inference takes 10 ms per $10,000$ points scan on an NVIDIA RTX 4060 Ti GPU. Furthermore, we assess the plug-and-play capability of POLI by integrating it into several existing LiDAR localization and mapping systems [16, 40]. Without any system-specific tuning or retraining, POLI yields substantial performance improvements across all evaluated pipelines, as illustrated in Table IV. These results suggest that POLI generalizes well across diverse robotic perception systems.

VI. Limitations

Although the proposed method achieves consistent improvements across multiple benchmarks, several limitations remain. First, our method faces challenges under out-of-distribution (OOD) scenarios. For example, when trained on HeLiPR [14] and tested on KITTI [8], POLI still improves downstream performance, but with smaller gains than in the in-domain setting. As shown in Table I, POLI substantially outperforms other augmentation baselines on HeLiPR [14], whereas its advantage becomes less pronounced on KITTI [8]. Second, our method relies on a local low-order geometric representation. A 3D Gaussian covariance captures only second-order moments—i.e., the first-fundamental-form aspects of the surface such as local anisotropy and tangential extent—and is therefore agnostic to the second fundamental form that encodes local surface curvature [7]. As a result, while our covariances faithfully describe dominant structures such as local planarity [26], they do not natively encode higher-order differential invariants (e.g., curvatures). Third, finding the global optimum of Eq. (13) in a differentiable manner remains challenging. A promising future direction is to develop a tighter relaxation of Eq. (13), for example, by exploiting Chordal Relaxation formulations.

VII. Conclusion

This paper revisits the role of point cloud geometry in robotic perception and highlights that a broad range of downstream tasks are fundamentally grounded in accurate estimation of underlying geometric structure. From this perspective, we introduced a unified theoretical and practical framework that models surface geometry as a statistical manifold and employs deep learning for principled estimation of point cloud geometry. Unlike prior learning-based methods and heuristic statistical approaches, our formulation is grounded in explicit mathematical principles that govern both geometric modeling and learning, thereby enabling self-supervised learning with strong geometric inductive biases.

As a practical implementation, we proposed POLI, a learning-based point cloud geometry estimator that directly predicts point-wise geometry in the form of Gaussian covariances. By enforcing geometric constraints during training in a principled manner, POLI learns a mapping between point cloud and their underlying geometry, producing consistent and informative geometric representations. Extensive experiments demonstrate that the proposed theory and practice successfully estimate the point cloud geometry, and consistently improve performance across a wide range of robotic perception tasks, where the learned geometry can be seamlessly integrated as 3D covariances, surface normals, or augmented point clouds. Overall, this work bridges geometric modeling and learning-based inference, offering a unified perspective on point cloud geometry representation and establishing statistical manifold-based covariance learning for geometric modeling as a promising paradigm for robust and scalable 3D robotic perception.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2023R1A2C2003130), and by the Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF) and the Unmanned Vehicle Advanced Research Center (UVARC), funded by the Ministry of Science and ICT, Republic of Korea (No. 2020M3C1C1A01082375).

References

[1] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2000. URL https://books.google.co.kr/books?hl=ko&lr=&id=vc2FWSo7wLUC&oi=fnd&pg=PR7&dq=Methods+of+information+geometry&ots=4JsyKx44L_&sig=ZMJpkAT36TZSpRPPkYM-gxyXLLM&redir_esc=y#v=onepage&q=Methods%20of%20information%20geometry&f=false.

[2] Yizhak Ben-Shabat and Stephen Gould. DeepFit: 3D Surface Fitting via Neural Network Weighted Least Squares. In *European Conference on Computer Vision (ECCV)*, 2020. doi: 10.1007/978-3-030-58452-8_2. URL https://link.springer.com/chapter/10.1007/978-3-030-58452-8_2.

[3] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. Nesti-Net: Normal Estimation for Unstructured 3D Point Clouds Using Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. doi: 10.1109/CVPR.2019.01035. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Ben-Shabat_Nesti-Net_Normal_Estimation_for_Unstructured_3D_Point_Clouds_Using_Convolutional_CVPR_2019_paper.html.

[4] Paul J. Besl and Neil D. McKay. A Method for Registration of 3D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. doi: 10.1109/34.121791. URL https://www.spiedigitallibrary.org/conference-proceedings-of-spie/1611/1/Method-for-registration-of-3-D-shapes/10.1117/12.57955.short.

[5] Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in neural information processing systems*, 35:5230–5242, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/228b9279ecf9bbafe582406850c57115-Abstract-Conference.html.

[6] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. URL https://dl.acm.org/doi/abs/10.1145/237170.237269.

[7] Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. URL https://ieeexplore.ieee.org/abstract/document/6248074.

[9] Michael Grupp. EVO: Python Package for the Evaluation of Odometry and SLAM, 2017. URL https://github.com/MichaelGrupp/evo.

[10] Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Min Lu, and Jianwei Wan. Rotational Projection Statistics for 3D Local Surface Description and Object Recognition. *International Journal of Computer Vision*, 105(1):63–86, 2013. doi: 10.1007/s11263-013-0627-y. URL https://link.springer.com/article/10.1007/s11263-013-0627-y.

[11] Connor Holmes, Frederike Dümbgen, and Timothy Barfoot. On semidefinite relaxations for matrix-weighted state-estimation problems in robotics. *IEEE Transactions on Robotics*, 40:4805–4824, 2024. URL https://ieeexplore.ieee.org/abstract/document/10706005.

[12] Connor Holmes, Frederike Dümbgen, and Timothy D Barfoot. SDPRLayers: Certifiable Backpropagation Through Polynomial Optimization Problems in Robotics. *IEEE Transactions on Robotics*, 2025. URL https://ieeexplore.ieee.org/abstract/document/11029144/.

[13] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, pages 71–78, 1992. URL https://dl.acm.org/doi/abs/10.1145/133994.134011.

[14] Minwoo Jung, Wooseong Yang, Dongjae Lee, Hyeonjae Gil, Giseop Kim, and Ayoung Kim. HeLiPR: Heterogeneous LiDAR Dataset for Inter-LiDAR Place Recognition Under Spatiotemporal Variations. *The International Journal of Robotics Research*, 43(3):1867–1883, 2024. URL https://journals.sagepub.com/doi/full/10.1177/02783649241242136.

[15] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Voxelized GICP for Fast and Accurate 3D Point Cloud Registration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 11054–11059, 2021. URL https://ieeexplore.ieee.org/abstract/document/9560835.

[16] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. GLIM: 3D range-inertial localization and mapping with GPU-accelerated scan matching factors. *Robotics and Autonomous Systems*, 179: 104750, 2024. URL https://www.sciencedirect.com/science/article/pii/S0921889024001349.

[17] Youngsun Kwon, Minhyuk Sung, and Sung-Eui Yoon. Implicit LiDAR network: LiDAR super-resolution via interpolation weight prediction. In *2022 international conference on robotics and automation (ICRA)*, pages 8424–8430. IEEE, 2022. URL https://ieeexplore.ieee.org/abstract/document/9811992.

[18] Daehan Lee, Hyungtae Lim, and Soohee Han. GenZ-ICP: Generalizable and Degeneracy-Robust LiDAR Odometry Using an Adaptive Weighting. *IEEE Robotics and Automation Letters*, 9(1):456–463, 2024. URL https://ieeexplore.ieee.org/abstract/document/10753079.

[19] Jan Eric Lenssen, Christian Osendorfer, and Jonathan Masci. Deep iterative surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11247–11256, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Lenssen_Deep_Iterative_Surface_Normal_Estimation_CVPR_2020_paper.html.

[20] Qing Li, Huifang Feng, Xun Gong, and Yu-Shen Liu. Learning Normals of Noisy Points by Local Gradient-Aware Surface Filtering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28828–28838, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Li_Learning_Normals_of_Noisy_Points_by_Local_Gradient-Aware_Surface_Filtering_ICCV_2025_paper.html.

[21] Yuanqi Li, Jingcheng Huang, Hongshen Wang, Peiyuan Lv, Yansong Liu, Jiuming Zheng, Jie Guo, and Yanwen Guo. High-quality Point Cloud Oriented Normal Estimation via Hybrid Angular and Euclidean Distance Encoding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1287–1296, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Li_High-quality_Point_Cloud_Oriented_Normal_Estimation_via_Hybrid_Angular_and_CVPR_2025_paper.html.

[22] Hyungtae Lim, Beomsoo Kim, Daebeom Kim, Eungchang Mason Lee, and Hyun Myung. Quatro++: Robust global registration exploiting ground segmentation for loop closing in LiDAR SLAM. *The International Journal of Robotics Research*, 43(5):685–715, 2024. URL https://journals.sagepub.com/doi/full/10.1177/02783649231207654.

[23] Hyungtae Lim, Daebeom Kim, Gunhee Shin, Jingnan Shi, Ignacio Vizzo, Hyun Myung, Jaesik Park, and Luca Carlone. KISS-Matcher: Fast and Robust Point Cloud Registration Revisited. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2025. URL https://ieeexplore.ieee.org/abstract/document/11127458.

[24] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. URL https://books.google.co.kr/books?hl=ko&lr=&id=AKuMj4PN_EMC&oi=fnd&pg=PR11&dq=Information+theory,+inference+and+learning+algorithms&ots=EOnrla3wBb&sig=6ziiQReq1O5rpGa9MGPyKNwGdeI&redir_esc=y#v=onepage&q=Information%20theory%2C%20inference%20and%20learning%20algorithms&f=false.

[25] Lucas Nunes, Rodrigo Marcuzzi, Benedikt Mersch, Jens Behley, and Cyrill Stachniss. Scaling Diffusion Models to Real-World 3D LiDAR Scene Completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14770–14780, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Nunes_Scaling_Diffusion_Models_to_Real-World_3D_LiDAR_Scene_Completion_CVPR_2024_paper.html.

[26] Mark Pauly, Richard Keiser, and Markus Gross. Multi-scale feature extraction on point-sampled surfaces. In *Computer Graphics Forum*, volume 22, pages 281–289. Wiley Online Library, 2003. URL https://onlinelibrary.wiley.com/doi/full/10.1111/1467-8659.00675.

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances*

in Neural Information Processing Systems, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d8bf84be3800d12f74d8b05e9b89836f-Abstract.html.

[28] Shi Qiu, Saeed Anwar, and Nick Barnes. Pu-transformer: Point cloud upsampling transformer. In *Proceedings of the Asian conference on computer vision*, pages 2475–2493, 2022. URL https://openaccess.thecvf.com/content/ACCV2022/html/Qiu_PU-Transformer_Point_Cloud_Upsampling_Transformer_ACCV_2022_paper.html.

[29] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, 2001. URL https://ieeexplore.ieee.org/abstract/document/924423.

[30] Radu Bogdan Rusu, Zoltan C. Marton, Nico Blodow, and Michael Beetz. Persistent Point Feature Histograms for 3D Point Clouds. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS)*, 2008. URL https://ebooks.iospress.nl/doi/10.3233/978-1-58603-887-8-119.

[31] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE, 2009.

[32] Avi Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems*, 2009. URL https://www.robots.ox.ac.uk/~avsegal/resources/papers/Generalized_ICP.pdf.

[33] Minkyun Seo, Hyungtae Lim, Kanghee Lee, Luca Carlone, and Jaesik Park. BUFFER-X: Towards Zero-Shot Point Cloud Registration in Diverse Scenes. *arXiv preprint arXiv:2503.07940*, 2025. URL https://arxiv.org/abs/2503.07940.

[34] Jingnan Shi, Heng Yang, and Luca Carlone. ROBIN: a graph-theoretic approach to reject outliers in robust estimation using invariants. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13820–13827. IEEE, 2021. URL https://ieeexplore.ieee.org/abstract/document/9562007.

[35] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique Signatures of Histograms for Local Surface Description. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 356–369, 2010. URL https://link.springer.com/chapter/10.1007/978-3-642-15558-1_26.

[36] Ignacio Vizzo, Benedikt Mersch, Rodrigo Marcuzzi, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Make it dense: Self-supervised geometric scan completion of sparse 3D lidar scans in large outdoor environments. *IEEE Robotics and Automation Letters*, 7 (3):8534–8541, 2022. URL https://ieeexplore.ieee.org/abstract/document/9812507.

[37] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters*, 8(2):1029–1036, 2023. URL https://ieeexplore.ieee.org/abstract/document/10015694/.

[38] Han Wang, Chen Wang, Chun-Lin Chen, and Lihua Xie. F-LOAM: Fast Lidar Odometry and Mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. URL https://ieeexplore.ieee.org/abstract/document/9636655.

[39] Wei Xu, Yixi Cai, Dawei He, Jiarong Lin, and Fu Zhang. FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE Robotics and Automation Letters*, 2021. doi: 10.1109/LRA.2021.3064227. URL https://ieeexplore.ieee.org/abstract/document/9372856/.

[40] Wei Xu, Yixi Cai, Dawei He, Jiarong Lin, and Fu Zhang. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Transactions on Robotics*, 2022. doi: 10.1109/TRO.2022.3141876. URL https://ieeexplore.ieee.org/abstract/document/9697912/.

[41] Bin Yang, Patrick Pfreundschuh, Roland Siegwart, Marco Hutter, Peyman Moghadam, and Vaishakh Patil. Tulip: Transformer for upsampling of lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15354–15364, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_TULIP_Transformer_for_Upsampling_of_LiDAR_Point_Clouds_CVPR_2024_paper.html.

[42] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020. URL https://ieeexplore.ieee.org/abstract/document/9286491/.

[43] 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)*, pages 11821–11830, 2020. doi: 10.1109/CVPR42600.2020.01184. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Yew_RPM-Net_Robust_Point_Matching_Using_Learned_Features_CVPR_2020_paper.html.

[44] Ji Zhang and Sanjiv Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014. doi: 10.15607/RSS.2014.X.007. URL https://www.ri.cmu.edu/pub_files/2014/7/Ji_LidarMapping_RSS2014_v8.pdf.

[45] Linqi Zhou, Yilun Du, and Jiajun Wu. 3D shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Zhou_3D_Shape_Generation_and_Completion_Through_Point-Voxel_Diffusion_ICCV_2021_paper.html?ref=https://githubhelp.com.

[46] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast Global Registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 766–

782, 2016. URL https://link.springer.com/chapter/10.1007/978-3-319-46475-6_47.

[47] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A Modern Library for 3D Data Processing. *arXiv preprint arXiv:1801.09847*, 2018. URL https://arxiv.org/abs/1801.09847.