IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

SVN-ICP: Uncertainty Estimation of ICP-based LiDAR Odometry using Stein Variational Newton

Shiping Ma[1†], *Student Member, IEEE*, Haoming Zhang[2†], *Member, IEEE*, and Marc Toussaint[1], *Member, IEEE*

***Abstract*—This letter introduces SVN-ICP, a novel Iterative Closest Point (ICP) algorithm with uncertainty estimation that leverages Stein Variational Newton (SVN) on manifold. Designed specifically for fusing LiDAR odometry in multisensor systems, the proposed method ensures accurate pose estimation and consistent noise parameter inference, even in LiDAR-degraded environments. By approximating the posterior distribution using particles within the Stein Variational Inference framework, SVN-ICP eliminates the need for explicit noise modeling or manual parameter tuning. To evaluate its effectiveness, we integrate SVN-ICP into a simple error-state Kalman filter alongside an IMU and test it across multiple datasets spanning diverse environments and robot types. Extensive experimental results demonstrate that our approach outperforms best-in-class methods on challenging scenarios while providing reliable uncertainty estimates. We release our code at https://github.com/LIS-TU-Berlin/SVN-ICP.git. A high-resolution video demonstration is available at https://youtu.be/c4QsMd1weik.**

***Index Terms*—SLAM, Probabilistic Inference, Sensor Fusion, ICP, Uncertainty Estimation**

I. Introduction

LOCALIZATION using light detection and ranging (LiDAR) sensors has become a cornerstone of robotic navigation over the past decades. Owing to their high ranging accuracy and robustness to varying lighting conditions, LiDAR sensors produce point clouds that can be exploited through scan matching techniques to estimate robot's pose.

In general, scan matching in LiDAR odometry (LO) can be performed at the raw point level (i.e. direct LO) or at the feature level (i.e. feature-based LO) [1]. Unlike the latter, direct LO is generally considered more generalizable and accurate at the cost of higher computational complexity, as it conducts scan matching on point level.

Taken the point-to-point ICP algorithm [2] as an example, scan matching aims to find the best transformation $\hat{\boldsymbol{T}} = \{\hat{\boldsymbol{R}},\ \hat{\boldsymbol{p}}\} \in SO(3) \times \mathbb{R}^3$ between two associated point clouds by solving

$$\hat{\boldsymbol{T}} \leftarrow \underset{\boldsymbol{T}}{\operatorname{argmin}} \underbrace{\sum_{n=1}^{N} \|\boldsymbol{q}_n - (\boldsymbol{R}\boldsymbol{p}_n + \boldsymbol{p})\|^2}_{\text{Loss function } \mathcal{L}(\boldsymbol{T})}, \tag{1}$$

where each point $\boldsymbol{p}_n$ in the source point cloud with N points can be transformed by $\hat{\boldsymbol{T}}$ to match its corresponding point $\boldsymbol{q}_n$ in the target point cloud.

This work was supported by the Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG).

†: Equally contributed authors.

[1]Shiping Ma and Marc Toussaint are with the Learning & Intelligent Systems Research Lab, Technical University of Berlin, Berlin, and with the Robotics Institute Germany (RIG), Germany.

[2]Haoming Zhang is with the Learning Systems and Robotics Lab, Technical University of Munich, Munich, and with the Munich Institute of Robotics and Machine Intelligence (MIRMI), Germany.

Corresponding: shiping.ma@tu-berlin.de and haoming.zhang@tum.de.

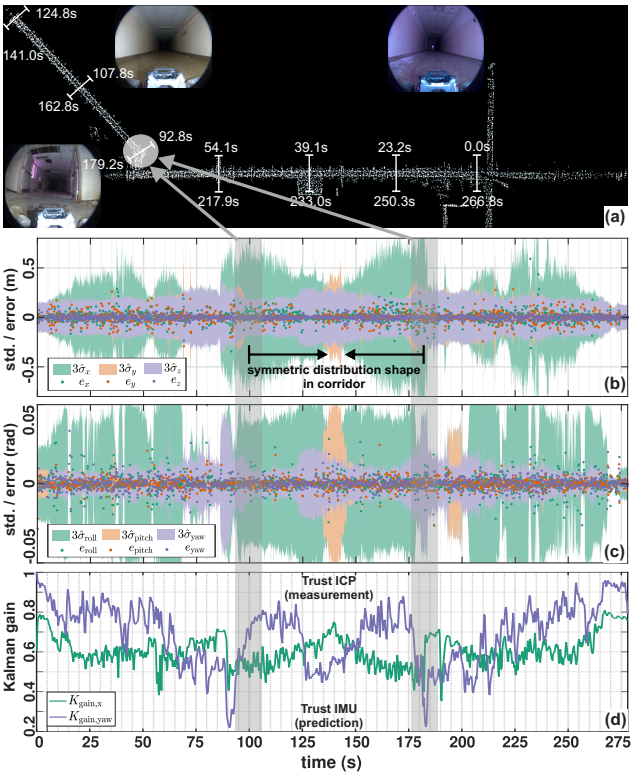

Figure 1: Uncertainty-aware LiDAR-Inertial Odometry using SVN-ICP (ours) evaluated on the `Long Corridor` sequence from the `SubT-MRS` dataset [7]. (a) Generated LiDAR map with timestamp annotations. (b-c) Estimated 6-D uncertainties (variance) visualized as $3\hat{\sigma}$ bounds along with error samples from the test run. (d) Smoothed Kalman gain computed given the estimated ICP variance.

However, since the optimization in (1) yields only a point estimate, it lacks the uncertainty characterization necessary for multisensor fusion. Many prior fusion methods rely on fixed heuristics [3] and hand-crafted noise models based on sensor noise, geometry, or scan convergence [4]. In contrast, prior studies [5, 6] highlight two key sources of uncertainty: (a) degenerative observations from noise or poor geometry, and (b) algorithmic issues like local minima or incorrect prior information. Beyond the fact that many of these sources cannot be sufficiently captured in advance, degraded environments, such as dust-filled caves, may exhibit a combination of multiple uncertainty sources, making uncertainty modeling a challenging problem.

To address this problem, Maken et al. [8] proposed Stein ICP, which advances the probabilistic ICP algorithm by leveraging Stein variational gradient descent (SVGD) [9] to enable principled uncertainty estimation. SVGD is well-known as a parallelizable and practical implementation of Stein variational inference that approximates the posterior distribution using a set of particles. In turn, a direct uncertainty measure can be

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

naturally derived from the non-parametric variational distribution without requiring any explicit uncertainty modeling or detection techniques.

However, Stein ICP still faces several limitations. As a first-order method, it inherits the drawbacks of gradient-based updates and performs poorly on ill-conditioned problems [10], particularly in environments with degenerate structures. This often results in slow, zig-zag convergence and high sensitivity to step size. Moreover, its Euclidean pose representation requires separate gradient computations and kernel designs for translation and rotation, which can deviate from the underlying $SE(3)$ manifold and hinder convergence.

Motivated by these observations, we propose SVN-ICP, a method tailored for LiDAR-based robot localization that extends previous approaches with two key innovations. First, it employs manifold-based state representation for computing scan-matching gradients using the right-hand perturbation model, eliminating the need for separately designing the kernel and computing the gradient for the translational and rotational parts. Second, it replaces SVGD with Stein Variational Newton (SVN) [11], incorporating second-order information (i.e. Hessian) to improve convergence stability and speed. Additionally, voxel-based point sampling and an early-stopping criterion further enhance efficiency and robustness.

Our experiments show that these innovations enhance both odometry and sensor fusion performance. Unlike [8], we thoroughly evaluate standalone LO and uncertainty estimates in a loosely coupled LiDAR–inertial system using lengthy real-world datasets, with both quantitative and qualitative analysis. Fig. 1 exemplifies the effectiveness of SVN-ICP's uncertainty estimation, validated through a Kalman filter, where the Kalman gain reliably reflects LiDAR degradation (e.g., in corridor-like environments).

Our contributions are threefold: 1) **Novel Algorithm:** We propose a novel ICP-based LiDAR odometry method, SVN-ICP, featuring built-in uncertainty quantification using Stein Variational Newton (SVN) on manifold, 2) **Comprehensive Evaluation:** We evaluate the estimated uncertainty by integrating SVN-ICP into a Kalman filter and benchmarking it against best-in-class methods on two challenging datasets, 3) **Implementation:** we develop an early-stopping mechanism for SVN and provide an efficient implementation in `C++` with `GPU` support.

II. Related Work

A. LiDAR-based Odometry

Decades of intensive research have shaped the current state of LiDAR-based odometry, which is primarily categorized into feature-based and direct methods [1].

Within this scope, feature-based LiDAR odometry relies on matching sparse, distinctive geometric features, such as edges and planar [12], or surface structures [13], and achieves high performance with efficient computation in structured environments. Another prominent class of methods performs scan matching directly at the raw point cloud level, typically building on the Iterative Closest Point (ICP) [2] or Normal Distribution Transform (NDT) [14] algorithms. Since ICP-based methods generally achieve higher accuracy [15], many recent studies have advanced this technique by leveraging temporal continuity in scan matching [16], adaptively adjusting thresholding parameters [17], or maintaining map representations effectively by considering structural information and estimation uncertainty [18].

Often, scan matching is integrated as a measurement update within a state estimator (e.g., a Kalman filter), with state propagation performed using an Inertial Measurement Unit (IMU), resulting in a LiDAR-inertial odometry system. Most notable approaches perform tightly coupled fusion using Kalman filters, where residuals of individual feature points are computed to correct the propagated state [4, 19], achieving both high accuracy and robustness.

However, since these methods primarily target nominal operating conditions, their performance generally degrades in challenging environments. To address this issue, many recent studies leverage additional incidental measurements from LiDAR, such as intensity [20, 21] and Doppler velocity [22], to enhance robustness in degraded scenarios.

Discussion: Despite their success in robot localization, existing methods do not explicitly address uncertainty awareness in degeneracy scenarios. Most rely on hand-crafted noise models or additional inputs, lacking reliable uncertainty estimates that are crucial for robust state estimation.

B. Uncertainty-Aware LiDAR Odometry

Recent studies have increasingly focused on incorporating uncertainty awareness into LiDAR-based localization, which can be broadly divided into degeneracy detection and uncertainty estimation.

Degeneracy (aka localizability) detection adopts a discriminative perspective by analyzing normal vectors [23], scan-matching residuals [24], or the Hessian matrix [25, 26], to down-weight degraded features or directions and reshape the covariance matrix for optimization. Recent approaches also extend this concept by incorporating multiple LiDARs [27] or additional sensing modalities [28]–[30] to better handle LiDAR degradation. Furthermore, learning-based methods for localizability detection in LiDAR point clouds have been investigated [31].

In contrast to degeneracy detection, the second line of work focuses on directly quantifying the uncertainty of ICP-based LiDAR odometry. While classical ICP methods using single value decomposition [32] or quaternions [2] can compute covariance matrices, many earlier studies have noted their tendency to be overoptimistic and proposed closed-form covariance estimations by explicitly modeling key sources of uncertainty [5, 6], or by adopting learning-based approaches [33, 34] to better capture the uncertainty in complex environments.

Another principled scan-matching approach with uncertainty estimation is based on the Bayesian formulation, exemplified by sampling-based Bayesian ICP [35] which uses a computation-intensive Markov Chain Monte Carlo (MCMC) algorithm to approximate the distribution of the estimated transformation. To improve efficiency, the same authors later proposed Stein ICP [8], which replaces MCMC with Stein

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

variational inference, significantly reducing computational cost while maintaining comparable accuracy.

Discussion: While methods based on degeneracy detection either require carefully tuned threshold parameters or rely on additional sensor inputs within complex frameworks, they still face challenges in threshold tuning and system adaptation across diverse environments. Other approaches that explicitly model or implicitly predict sources of uncertainty may struggle to generalize across scenarios, often resulting in overconfident uncertainty estimates.

These limitations motivate our investigation of uncertainty estimation for LiDAR odometry using Stein variational inference, a one-shot method to approximate the ICP posterior that avoids extensive tuning and over-engineering.

III. Preliminaries

A. Stein Variational Gradient Descent

Stein's method [9] establishes a probabilistic distance measure between the posterior (target) distribution $p(\boldsymbol{\xi})$ and a latent distribution $q(\boldsymbol{\xi})$ by computing the kernelized Stein discrepancy

$$\mathbb{S}(q,p) = \max_{\boldsymbol{\phi}\in\mathcal{H}^d} \left\{ [\mathbb{E}_{\boldsymbol{\xi}\sim q}\mathrm{Tr}(\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{\xi}))]^2,\ s.t.\ \|\boldsymbol{\phi}\|_{\mathcal{H}^d} \le 1 \right\} \tag{2}$$

using a vector-valued smooth function $\boldsymbol{\phi}(\boldsymbol{\xi}) = [\phi_1(\boldsymbol{\xi}), \cdots, \phi_d(\boldsymbol{\xi})]$ from a reproducing kernel Hilbert space (RKHS) $\mathcal{H}^d$. The function $\mathrm{Tr}(\cdot)$ computes the matrix trace. And the function $\boldsymbol{\phi}(\boldsymbol{\xi})$ must satisfy the Stein identity, i.e., $\mathbb{E}_{\boldsymbol{\xi}\sim p}[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{\xi})] = 0$, where the Stein operator is defined as

$$\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{\xi}) = \nabla_{\boldsymbol{\xi}} \log p(\boldsymbol{\xi})^T + \nabla_{\boldsymbol{\xi}}\boldsymbol{\phi}(\boldsymbol{\xi}). \tag{3}$$

In particular, the quantity $\mathbb{E}_{\boldsymbol{\xi}\sim q}[\mathrm{Tr}(\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{\xi}))]^2$ serves as a discrepancy metric that equals zero if and only if $q = p$.

Taken the Stein discrepancy as a variational objective, SVGD [9] employs a set of K particles $\{\boldsymbol{\xi}_k\}_k^K$ to represent q, updated with a small perturbation $\epsilon\boldsymbol{\phi}(\boldsymbol{\xi}_k)$ as

$$\boldsymbol{\xi}_k \leftarrow \boldsymbol{\xi}_k + \epsilon\boldsymbol{\phi}(\boldsymbol{\xi}_k), \quad k = 1, \ldots, K, \tag{4}$$

which gradually drives the particles to match the target distribution p. The optimal vector function $\boldsymbol{\phi}^*$, in the limit as the step size $\epsilon \to 0$, corresponds to the steepest descent direction of the Kullback-Leibler (KL) divergence, which gives us

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \left\{ -\frac{\mathrm{d}}{\mathrm{d}\epsilon} KL(q_{\boldsymbol{\phi}} \,\|\, p)\big|_{\epsilon=0} \right\}. \tag{5}$$

As all values in the function $\boldsymbol{\phi}$ in (2) lie within the unit ball $\mathcal{B} = \{\boldsymbol{\phi} \in \mathcal{H}^d : \ \|\boldsymbol{\phi}\|_{\mathcal{H}^d} \le 1\}$ of $\mathcal{H}^d$ in the vector-valued RKHS, defined by the positive definite kernel $k(\boldsymbol{\xi}, \cdot)$, we can exploit the reproducing property $\mathbb{E}_q[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{\xi})] = \langle \boldsymbol{\phi},\ \mathbb{E}_q[\mathcal{A}_p k(\boldsymbol{\xi}, \cdot))] \rangle$ to compute (5). By substituting (3) in (5), we have

$$\boldsymbol{\phi}^* = \mathbb{E}_{\boldsymbol{\xi}\sim q}\mathrm{Tr}(\mathcal{A}_p k(\boldsymbol{\xi}, \cdot)) \tag{6}$$

$$= \mathbb{E}_{\boldsymbol{\xi}\sim q}[\underbrace{\nabla_{\boldsymbol{\xi}_i} \log p(\boldsymbol{\xi}_i) k(\boldsymbol{\xi}, \cdot)}_{\text{Steepest descent direction}} + \underbrace{\nabla_{\boldsymbol{\xi}_i} k(\boldsymbol{\xi}, \cdot)}_{\text{Repulsion force}}], \tag{7}$$

where the first term directs particles toward regions of high probability under $p(\boldsymbol{\xi}_i)$, and the second term introduces a repulsive force that discourages particles from collapsing around a single mode, thereby promoting spread across the support of $p(\boldsymbol{\xi}_i)$.

B. Stein ICP

Stein ICP [8] follows the concept of the SGD-ICP [36], which solves the ICP problem in (1) using stochastic gradient descent [37]. In contrast to standard ICP solvers, SGD-ICP partitions the dense point cloud into mini-batches $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_1 \ldots \mathcal{M}_M\}$ and optimizes the transformation $\hat{\boldsymbol{T}}_{ij}$ by iteratively updating the state parameters, similarly to (4). However, unlike (4), the gradient $\bar{g}$ of the state parameter $\boldsymbol{\xi}$ must be separately computed for the translational and the rotational parts (see Equation (4)-(5) in [8]).

Built on SGD-ICP, Stein ICP leverages SVGD to compute gradients of the state parameters in (4) by adapting (7) as

$$\boldsymbol{\phi}^* = \sum_{k=1}^{K} \Big[-\underbrace{\big(N\bar{g}(\boldsymbol{\xi}_k^t, \mathcal{M}^t) + \nabla \log p_0(\boldsymbol{\xi}_k)\big)}_{\nabla_{\boldsymbol{\xi}_k} \log p(\boldsymbol{\xi}_k)} k(\boldsymbol{\xi}_k, \boldsymbol{\xi}) + \nabla_{\boldsymbol{\xi}_k} k(\boldsymbol{\xi}_k, \boldsymbol{\xi}) \Big], \tag{8}$$

where the gradient of the posterior log-density $\nabla_{\boldsymbol{\xi}_k} \log p(\boldsymbol{\xi}_k)$ consists of two components: the mini-batch gradients computed, and the gradient of the log prior distribution (see Equation (14)-(15) in [8]). An RBF kernel with a modified distance metric is used.

IV. SVN-ICP

To overcome the limitations of Stein ICP introduced in Sec. I, we propose SVN-ICP, a frame-to-map LiDAR odometry method with uncertainty estimation that improves upon Stein ICP in both accuracy and convergence speed.

A. ICP on Manifold

In this work, we consider solving the point-to-point scan matching problem in (1) by formulating the state transformation on a manifold using the following loss function:

$$\mathcal{L}(\boldsymbol{\xi}) = \sum_{n=1}^{N} \| \underbrace{(\exp(\boldsymbol{\vartheta})\boldsymbol{p}_n + \boldsymbol{p}) - \boldsymbol{q}_n}_{\boldsymbol{e}_n} \|^2, \tag{9}$$

where the vector $\boldsymbol{\xi} = [\boldsymbol{p}\ \boldsymbol{\vartheta}]^T = [x\ y\ z\ \theta_x\ \theta_y\ \theta_z]^T \in \mathbb{R}^3 \times \mathfrak{so}(3)$ denotes the minimal pose increment between two point clouds. To enable iterative optimization, we introduce the right-hand perturbation state $\Delta\boldsymbol{\xi}$ around a nominal state $\bar{\boldsymbol{\xi}}_{\mathrm{op}}$ in (9). This yields the perturbed loss function:

$$\mathcal{L}(\bar{\boldsymbol{\xi}}_{\mathrm{op}} \boxplus \Delta\boldsymbol{\xi}) = \sum_{n=1}^{N} \boldsymbol{e}_n(\bar{\boldsymbol{\xi}}_{\mathrm{op}} \boxplus \Delta\boldsymbol{\xi})^T \boldsymbol{e}_n(\bar{\boldsymbol{\xi}}_{\mathrm{op}} \boxplus \Delta\boldsymbol{\xi}), \tag{10}$$

where $\boxplus$ denotes the pose update on manifold [38].

As SVN-ICP follows Newton's method to optimize (10), we approximate the loss function using the second-order Taylor expansion as

$$\mathcal{L}(\bar{\boldsymbol{\xi}}_{\mathrm{op}} \boxplus \Delta\boldsymbol{\xi}) \approx \mathcal{L}(\bar{\boldsymbol{\xi}}_{\mathrm{op}}) + 2\boldsymbol{b}^T\Delta\boldsymbol{\xi} + \Delta\boldsymbol{\xi}^T\boldsymbol{H}\Delta\boldsymbol{\xi}, \tag{11}$$

where the gradient $\boldsymbol{b}$ and the approximated Hessian $\boldsymbol{H}$ are given by

$$\boldsymbol{b} = \sum_{n=1}^{N} \boldsymbol{b}_n, \quad \boldsymbol{H} = \sum_{n=1}^{N} \boldsymbol{H}_n. \tag{12}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

The Jacobian $\boldsymbol{J}_n$, gradient $\boldsymbol{b}_n$, and Hessian $\boldsymbol{H}_n$ of the point $\boldsymbol{p}_n$ in the ICP process are computed as

$$\begin{aligned} \boldsymbol{J}_n &= \frac{\partial \boldsymbol{e}_n}{\partial \boldsymbol{\xi}} = \begin{bmatrix} -[\boldsymbol{R}\boldsymbol{p}_n]_\times & \boldsymbol{R} \end{bmatrix} \in \mathbb{R}^{3\times 6}, \\ \boldsymbol{b}_n &= -\boldsymbol{J}_n^T \boldsymbol{e}_n, \quad \boldsymbol{H}_n = \boldsymbol{J}_n^T \boldsymbol{J}_n, \end{aligned} \tag{13}$$

where the operator $[\cdot]_\times$ converts the 3-D vector into its corresponding skew-symmetric matrix.

In this scheme, the optimal perturbation state $\Delta\boldsymbol{\xi}$ is obtained by solving the linear system $\boldsymbol{H}\Delta\boldsymbol{\xi} = \boldsymbol{b}$, which forms the estimated transformation given as

$$\hat{\boldsymbol{T}} = \bar{\boldsymbol{T}}_{\text{op}} \exp(\Delta\boldsymbol{\xi}). \tag{14}$$

Unlike prior works that use a left-hand perturbation model [4, 17, 24], we adopt a right-hand formulation that better decouples translation and rotation. This mitigates the coupling effects inherent in left-multiplicative updates and improves numerical stability in frame-to-map ICP, especially under large initial translations or when handling distant points, conditions frequently seen in odometry applications.

B. Solving ICP with Stein Variational Newton

We employ variational inference within the probabilistic ICP framework using the Stein Variational Newton (SVN) method [11]. In contrast to SVGD, which follows the gradient flow of the KL divergence (cf. (5)), SVN incorporates second-order information of the objective function (10) to compute Newton-like updates in function space.

Concretely, SVN seeks a Newton-like update direction under the approximation

$$-\frac{\mathrm{d}}{\mathrm{d}\epsilon} KL(q_{\boldsymbol{\phi}} \| p)\big|_{\epsilon=0} \approx \frac{\mathrm{d}^2}{\mathrm{d}\epsilon^2} KL(q_{[\boldsymbol{\phi},\boldsymbol{\theta}]} \| p)\big|_{\epsilon=0}, \tag{15}$$

where $\boldsymbol{\phi}$ is an arbitrary first-order functional perturbation (as in (5)) and $\boldsymbol{\theta}$ denotes an independent second-order perturbation. Accordingly, we define the Stein variational update of k-th particle with step size $\epsilon = 1$ as in [11] as

$$\boldsymbol{\xi}_k \leftarrow \boldsymbol{\xi}_k + \epsilon \boldsymbol{\theta}^*(\boldsymbol{\xi}_k), \quad \boldsymbol{\theta}^* = \tilde{\boldsymbol{H}}^{-1} \boldsymbol{\phi}^* \tag{16}$$

with the optimal Newton direction $\boldsymbol{\theta}^*$. Following Stein ICP (Sec. III-B), we compute the SVGD update $\boldsymbol{\phi}^*$ in (7) as

$$\boldsymbol{\phi}^*(\boldsymbol{\xi}_k) = \frac{1}{K} \sum_{l=1}^{K} \left[k(\boldsymbol{\xi}_l, \boldsymbol{\xi}_k) \boldsymbol{b}(\boldsymbol{\xi}_k) + \nabla_{\boldsymbol{\xi}_l} k(\boldsymbol{\xi}_l, \boldsymbol{\xi}_k) \right]. \tag{17}$$

The matrix-preconditioned Hessian $\tilde{\boldsymbol{H}}$ of SVN associated with the particle $\boldsymbol{\xi}_k$ is given by

$$\begin{aligned} \tilde{\boldsymbol{H}}(\boldsymbol{\xi}_k) &= \mathbb{E}_{\boldsymbol{\xi}\sim q} \left[\boldsymbol{H}_k(\boldsymbol{\xi}) k(\boldsymbol{\xi}, \boldsymbol{\xi}_k)^2 + (\nabla_{\boldsymbol{\xi}} k(\boldsymbol{\xi}, \boldsymbol{\xi}_k))^{\otimes 2} \right] \\ &= \frac{1}{K} \sum_{l=1}^{K} \left[\boldsymbol{H}_k(\boldsymbol{\xi}) k(\boldsymbol{\xi}_l, \boldsymbol{\xi}_k)^2 + (\nabla_{\boldsymbol{\xi}} k(\boldsymbol{\xi}_l, \boldsymbol{\xi}_k))^{\otimes 2} \right], \end{aligned} \tag{18}$$

where the repulsive force is computed using the operation $\boldsymbol{w}^{\otimes 2} = \boldsymbol{w}\boldsymbol{w}^\top$.

In (17) and (18), we use the RBF kernel $k(\boldsymbol{\xi}_l, \boldsymbol{\xi}_k) = \exp\left(\frac{1}{h}\|\boldsymbol{\xi}_l - \boldsymbol{\xi}_k\|_2^2\right)$ with bandwidth h in Stein's methods following [8].

C. The SVN-ICP Algorithm

Algorithm 1 outlines the pipeline of SVN-ICP (odometry only). Starting from a prior pose $\check{\boldsymbol{T}}$ with covariance $\check{\boldsymbol{\Sigma}}$, we sample K particles using a predefined variance σ to align the source point cloud $\boldsymbol{\mathcal{P}}$ to the target $\boldsymbol{\mathcal{Q}}$. To account for incremental LiDAR pose changes in odometry, we downsample $\boldsymbol{\mathcal{Q}}$ to a sub-target point cloud $\boldsymbol{\mathcal{Q}}_n$ via the KNN algorithm, retaining only neighbors of points in $\boldsymbol{\mathcal{P}}$. Unlike the mini-batch sampling in Stein ICP, our method ensures higher variational inference quality while significantly reducing computational cost.

After each SVN iteration, we compute the average norm of state updates across all particles. If it falls below the threshold $\boldsymbol{\epsilon}$, the optimization stops early (line 19–21). The final pose is given by the mean perturbation across particles (see (14)), while the estimated covariance is transformed to the global frame using the Adjoint of the prior pose, $\mathrm{Ad}_{\check{\boldsymbol{T}}}$ [39].

Algorithm 1: SVN-ICP

Input : Prior pose $\check{\boldsymbol{T}}$ and covariance $\check{\boldsymbol{\Sigma}}$,
Source cloud $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_n\},\ n = 1 \dots N_p$,
Target cloud $\boldsymbol{\mathcal{Q}} = \{\boldsymbol{q}_n\},\ n = 1 \dots N_q$.
Output: Posterior pose $\hat{\boldsymbol{T}}$ and covariance $\hat{\boldsymbol{\Sigma}}$

/* initialize K perturbation particles */
1 $\boldsymbol{\Xi} = \{\boldsymbol{\xi}_k\}_1^K$, where $\boldsymbol{\xi}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma})$;
/* find corresponding sub target point cloud $\boldsymbol{\mathcal{Q}}_n$ */
2 **parallelized for** each point $\boldsymbol{p}_n \in \boldsymbol{\mathcal{P}}$
3 $\quad \boldsymbol{\mathcal{Q}}_n \leftarrow \mathrm{KNN}(\boldsymbol{p}_n, \boldsymbol{\mathcal{Q}})$;
4 **end**
5 **for** *iteration* $i = 1 \dots I$ **do**
6 $\quad$ **parallelized for** each particle $\boldsymbol{\xi}_k \in \boldsymbol{\Xi}$
$\quad\quad$ /* update pose in new iteration */
7 $\quad\quad \boldsymbol{T}_{k,i} = \{\boldsymbol{R}_{k,i},\ \boldsymbol{p}_{k,i}\} = \check{\boldsymbol{T}}\boldsymbol{T}_{k,i-1}$;
8 $\quad\quad$ **parallelized for** each point $\boldsymbol{p}_n \in \boldsymbol{\mathcal{P}}$
9 $\quad\quad\quad \boldsymbol{p}'_n = \boldsymbol{R}_{k,i}\boldsymbol{p}_n + \boldsymbol{p}_{k,i}$; // transform point
10 $\quad\quad\quad \boldsymbol{q}_n \leftarrow \mathrm{KNN}(\boldsymbol{p}'_n, \boldsymbol{\mathcal{Q}}_n)$; // find correspond
11 $\quad\quad\quad$ Compute point residual $\boldsymbol{e}_{n,k} = \boldsymbol{p}'_n - \boldsymbol{q}_n$;
12 $\quad\quad\quad$ Compute $\boldsymbol{J}_{n,k}, \boldsymbol{b}_{n,k}, \boldsymbol{H}_{n,k}$; // Equation (13)
13 $\quad\quad$ **end**
14 $\quad\quad$ Compute $\boldsymbol{b}_{k,i}$ and $\boldsymbol{H}_{k,i}$; // Equation (12)
15 $\quad\quad$ Compute SVN Hessian $\tilde{\boldsymbol{H}}_{k,i}$; // Equation (18)
16 $\quad\quad$ Compute state update $\Delta\boldsymbol{\xi}_{k,i} = \tilde{\boldsymbol{H}}_{k,i}^{-1}\boldsymbol{\phi}^*_{k,i}$;
17 $\quad\quad$ Update particle pose $\boldsymbol{T}_{k,i} = \hat{\boldsymbol{T}}_k \exp(\Delta\boldsymbol{\xi}_k)$;
18 $\quad$ **end**
19 $\quad$ **if** $\frac{1}{K}\sum_{k=1}^{K} \|\Delta\boldsymbol{\xi}_k\|^2 < \boldsymbol{\epsilon}$ **then**
20 $\quad\quad$ break ; // early-stop condition reached
21 $\quad$ **end if**
22 **end for**
/* compute mean perturbation state and covariance */
23 $\bar{\boldsymbol{\xi}}_{\text{icp}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\xi}_k$, $\bar{\boldsymbol{\Sigma}}_{\text{icp}} = \frac{1}{K}\sum_{k=1}^{K} (\boldsymbol{\xi}_k - \bar{\boldsymbol{\xi}})(\boldsymbol{\xi}_k - \bar{\boldsymbol{\xi}})^T$
/* compute final pose and propagate covariance */
24 $\hat{\boldsymbol{T}} = \check{\boldsymbol{T}} \exp(\bar{\boldsymbol{\xi}}_{\text{icp}})$, $\hat{\boldsymbol{\Sigma}} = \check{\boldsymbol{\Sigma}} + \mathrm{Ad}_{\check{\boldsymbol{T}}} \bar{\boldsymbol{\Sigma}}_{\text{icp}} \mathrm{Ad}_{\check{\boldsymbol{T}}}^T$
25 **return** $\hat{\boldsymbol{T}}$, $\hat{\boldsymbol{\Sigma}}$

V. Implementation

We evaluate SVN-ICP in two settings: standalone LiDAR odometry (LO) for baseline performance validation, and a loosely coupled LiDAR–Inertial odometry (LIO) system using an error-state Kalman filter. We assume that the state propagation in the standalone LO follows a constant-velocity motion model, as also utilized in [8, 17]. In the LIO setup, the LiDAR noise parameter is either fixed or adaptively updated using

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

SVN-ICP's estimated covariance, demonstrating the benefit of its uncertainty estimation.

The Kalman filter state and its error-state vector are defined at the IMU center as

$$\boldsymbol{x}_b^w \coloneqq \{\boldsymbol{T}_b^w,\ \boldsymbol{v}_b^w,\ \boldsymbol{b}_{\mathrm{acc}},\ \boldsymbol{b}_{\mathrm{gyro}}\} \in SE(3) \times \mathbb{R}^9, \tag{19}$$

$$\Delta\boldsymbol{x} \coloneqq [\Delta\boldsymbol{p}^T\ \Delta\boldsymbol{v}^T\ \Delta\boldsymbol{\vartheta}^T\ \Delta\boldsymbol{b}_{\mathrm{acc}}^T\ \Delta\boldsymbol{b}_{\mathrm{gyro}}^T]^T \in \mathbb{R}^{15}, \tag{20}$$

where $\boldsymbol{T}_b^w = \{\boldsymbol{R}_b^w\ \boldsymbol{p}_b^w\}$ and $\boldsymbol{v}_b^w$ denote the pose and velocity of the body frame $(\cdot)_b$ with respect to the world frame $(\cdot)^w$. The vectors $\boldsymbol{b}_{\mathrm{acc}}$ and $\boldsymbol{b}_{\mathrm{gyro}}$ represent the accelerometer and gyroscope biases. The orientation error $\Delta\boldsymbol{\vartheta} = [\Delta\theta_x\ \Delta\theta_y\ \Delta\theta_z]^T$ is expressed in axis-angle form.

We adopt the same IMU mechanization from [4] to propagate the state $\check{\boldsymbol{T}}$ and covariance $\check{\boldsymbol{\Sigma}}$. Since SVN-ICP provides the optimal perturbation state (i.e., Kalman innovation) $\bar{\boldsymbol{\xi}}$ and its covariance $\bar{\boldsymbol{\Sigma}}_{\mathrm{icp}}$ (line 23 of Alg. 1), the error-state vector is computed with Kalman gain $\boldsymbol{K}_{\mathrm{g}}$ as follows

$$\Delta\boldsymbol{x} = \boldsymbol{K}_{\mathrm{g}}\bar{\boldsymbol{\xi}}, \quad \boldsymbol{K}_{\mathrm{g}} = \check{\boldsymbol{\Sigma}}\boldsymbol{C}^T \cdot (\boldsymbol{C}\check{\boldsymbol{\Sigma}}\boldsymbol{C}^T + \bar{\boldsymbol{\Sigma}}_{\mathrm{icp}})^{-1}, \tag{21}$$

where the observation matrix $\boldsymbol{C} \in \mathbb{R}^{6\times 15}$ is defined as

$$\boldsymbol{C} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{22}$$

We implemented our approach in `C++`, using sensor interfaces from the Robot Operating System `ROS2` [40]. GPU-based data handling and basic math operations were supported by `LibTorch` [41].

VI. Experiments and Results

To evaluate both odometry performance and the estimated uncertainty, we use two public datasets: *SubT-MRS* [7] and *GEODE* [42], which encompass representative LiDAR-degraded scenarios. While the *SubT-MRS* dataset primarily features mixed indoor environments, the *GEODE* dataset additionally includes challenging outdoor scenarios such as off-road trails and waterways.

We have re-implemented[1] the Stein ICP based on its original implementation[2]. Since the original Stein ICP performs frame-to-frame matching, we provide an enhanced version, SVGD-ICP. This enhanced variant follows the same odometry pipeline as SVN-ICP but employs SVGD as the optimization scheme. All experiments are conducted on a desktop equipped with an Intel i7-8086K CPU, 32 GB of RAM, and an NVIDIA GTX 1080 Ti GPU.

A. General Performance Metrics

General error metrics, including the Absolute Pose Error (APE) and Relative Pose Error (RPE), are computed using the `evo` package [45].

1) Results on `SubT-MRS` Dataset: We compare SVN-ICP in different settings against classical baselines. Table I presents the performance metrics on the `SubT-MRS` dataset.

While most well-engineered systems with loop-closure constraints, originally developed for competition purposes, achieve the highest accuracy on several sequences, our loosely coupled Kalman filter, leveraging the estimated ICP uncertainty, delivers competitive results without additional observations (e.g., loop-closure constraints). In contrast to other LiDAR-only odometry methods, which often fail in challenging scenarios, the proposed ICP variant optimized via both SVGD and SVN achieves competitive accuracy and demonstrates strong robustness across all test sequences. By comparison, the original Stein ICP [8], designed for frame-to-frame ICP using mini-batch sampling, is not suitable for odometry applications in such demanding environments.

When comparing SVGD-ICP and SVN-ICP, we observe that both methods exhibit similar performance in well-structured yet geometrically degraded environments. This suggests that SVGD, when solved using the Adam optimizer [46], provides sufficiently reliable descent directions for effective convergence. However, in mixed-degeneration environments, SVN-ICP consistently outperforms SVGD-ICP, highlighting the advantage of incorporating curvature information and stable Hessian-based updates in SVN-ICP, features that are better suited to handling degenerate conditions (more details in Sec. VI-C1).

When comparing LiDAR-only and LiDAR–Inertial Odometry methods, their performance differences are highly dependent on motion characteristics in our experiments. LiDAR-only approaches can achieve comparable performance in scenarios involving ground vehicles (e.g., UGVs or RC cars) with smooth motion. However, under aggressive motions, encountered with drones or hand-held devices, integrating IMU measurements in a fusion framework significantly enhances the stability and accuracy of the odometry.

Finally, our key result highlights the importance of consistent uncertainty parameterization in sensor fusion. While the Kalman filter configured with fixed ICP noise parameters[3] fails to produce reliable odometry estimates in several sequences (marked in red), dynamically updating the ICP noise using the proposed SVN-ICP leads to accurate and robust state estimation across all tested scenarios.

2) Results on `GEODE` Dataset: Consistent results as in Sec. VI-A1 are also observed on the `GEODE` dataset, as shown in Table II. However, since the test scenarios are less challenging compared to those in the `SubT-MRS` dataset, the noise-updated Kalman filter does not yield significant improvements, except in the `Waterway Long` sequence.

Surprisingly, all LiDAR–Inertial approaches based on Kalman filtering fail to outperform LiDAR-only methods (e.g., [24]) on the `Uneven Tunnel5` sequence. This outcome is likely due to high IMU noise caused by the uneven, under-construction tunnel floor. The `Bridge1` sequence demonstrates the most challenging scenario, where all benchmark methods fail due to its featureless, repetitive structure combined with high-speed motion. Although the accuracy of our proposed methods is not satisfactory in this case, they maintain stability and do not diverge, unlike the baselines, even in such adverse conditions.

Discussion: Based on the experimental observations above, we have successfully demonstrated effective uncertainty es-

[1][Online] https://github.com/msp666/SteinICP-Odometry-and-Mapping.

[2][Online] https://bitbucket.org/fafz/stein-icp.

[3]We adopt commonly used noise parameters for LiDAR odometry: $\boldsymbol{\eta}_{\mathrm{ICP,p}} \sim \mathcal{N}(\mathbf{0}, 1e^{-4})$ and $\boldsymbol{\eta}_{\mathrm{ICP,rot}} \sim \mathcal{N}(\mathbf{0}, 1e^{-5})$.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Table I: General performance metrics on the `SubT-MRS` dataset [7], with accuracy reported in APE / RPE (m) format. **RunTime** indicates the average processing time per LiDAR scan in seconds.

Scene / Robot Type		Geometric Degradation (UGV)					Simulation (Drone)			Mix Degradation				
	Method	Urban (1680.2 s)	Tunnel (3425.4 s)	Cave (1786.1 s)	Nuclear1 (513.0 s)	Nuclear2 (3177.0 s)	Factory (160.6 s)	Ocean (127.4 s)	Sewerage (130.9 s)	Laurel Caverns (H) (955.3 s)	Long Corridor (R) (280.1 s)	Multi Floor (L) (417.6 s)	Average	RunTime
system	Liu et al. [19]*	0.307 / 0.038	0.095 / 0.032	0.629 / 0.055	0.122 / 0.028	0.235 / 0.048	**0.889** / 0.191	0.757 / 0.174	**0.978** / 0.188	1.454 / 0.088	1.454 / 0.088	**0.401** / 0.059	**0.588** / 0.091	51.31
	Weitong et al. [43]*	**0.260** / 0.038	0.096 / 0.032	0.617 / 0.056	0.120 / 0.029	**0.222** / 0.049	0.998 / 0.190	0.770 / 0.183	1.586 / 0.243	0.402 / **0.046**	1.254 / 0.086	0.577 / **0.054**	0.663 / 0.097	0.125
	Kim et al [19]*	0.331 / 0.098	0.092 / 0.032	0.787 / 0.055	0.123 / 0.028	0.270 / 0.054	10.63 / 0.861	22.43 / 0.535	7.147 / 0.401	**0.279** / **0.046**	2.100 / 0.093	0.650 / 0.260	3.825 / 0.219	0.268
	Zhong et al. [44]*	1.205 / 0.157	0.695 / 0.062	- / -	1.175 / 0.079	1.720 / 0.106	**0.889** / 0.706	0.778 / 0.691	1.130 / 0.617	2.080 / 0.094	- / -	- / -	- / -	0.027
odom	LIO-EKF [4]*	1.060 / 0.130	0.220 / 0.090	0.750 / 0.150	0.470 / 0.130	0.620 / 0.200	4.920 / **0.040**	0.280 / 0.040	24.46 / 0.160	9.140 / 0.200	2.990 / 0.630	5.500 / 0.280	4.312 / 0.186	0.006
	KISS-ICP [17]	13.16 / 0.108	7.739 / 0.111	26.90 / 0.249	0.143 / 0.044	31.13 / 0.241	- / -	- / -	- / -	31.57 / 0.215	6.204 / 0.255	15.41 / 0.214	- / -	0.005
	GenZ-ICP [24]	0.600 / 0.023	0.101 / 0.021	0.564 / 0.037	0.555 / 0.033	- / -	- / -	- / -	- / -	0.399 / 0.050	1.746 / 0.073	20.49 / 0.101	- / -	0.042
	Stein ICP [8]	23.26 / 0.057	- / -	- / -	- / -	- / -	- / -	- / -	- / -	- / -	- / -	- / -	- / -	0.935
odom, ours	SVGD-ICP	0.562 / **0.022**	**0.077** / **0.018**	0.764 / **0.033**	0.141 / 0.024	- / -	- / -	22.57 / 1.050	20.88 / 0.570	16.79 / 0.067	1.334 / 0.080	14.89 / 0.092	- / -	0.669
	SVN-ICP	0.551 / 0.031	0.110 / 0.028	0.562 / 0.042	0.131 / 0.026	- / -	- / -	21.97 / 0.869	20.42 / 0.575	8.911 / 0.085	1.321 / **0.076**	11.15 / 0.088	- / -	0.483
	SVN-ICP+KF-fix	2.365 / 0.025	0.081 / 0.028	5.260 / 0.037	**0.058** / **0.017**	0.464 / **0.027**	4.570 / 0.137	1.173 / 0.033	11.51 / 0.107	5.197 / 0.056	1.567 / 0.086	0.692 / 0.054	2.995 / 0.055	0.568
	SVN-ICP+KF	0.478 / 0.030	0.085 / 0.027	**0.532** / 0.040	0.059 / 0.023	0.441 / 0.046	4.836 / 0.053	**0.047** / **0.031**	2.933 / **0.068**	4.089 / 0.056	**0.650** / 0.089	0.911 / 0.053	1.369 / 0.046	0.515

[1] Best results are shown in **bold**. The best among odometry-only methods are highlighted in blue, while red indicates notable differences when using fixed noise parameters (SVN-ICP+KF-fix).
[2] Results indicated by "*" are reported as presented in [7].
[3] Methods labeled as "system" are cascaded pipelines with multiple localization approaches, often fine-tuned with loop-closure constraints for competition purposes [7].
[4] Methods labeled as "odom" refer to pure LiDAR or LiDAR–Inertial odometry without loop closure and sophisticated engineering.
[5] SVGD-ICP adopts the SVN-ICP pipeline but replaces the descent direction with SVGD instead of second-order SVN.
[6] H, R, and L denote hand-held, RC car, and legged robot platforms, respectively. We excluded the Block LiDAR sequence from our experiments due to the absence of ground-truth data.
[7] We omitted results where the algorithm failed across multiple test runs or produced a large APE (e.g., > 100 m); these cases are denoted as "–".
[8] By default, Stein ICP, SVGD-ICP, and SVN-ICP use 30 particles optimized over a maximum of 100 iterations. All other parameters are kept identical to ensure a fair benchmark across all experiments.

Table II: APE (m) on the `GEODE` dataset [42].

Robot Type		UGV		Sailboat		Handheld	Vehicle
	Method	Offroad1 (426.6 s)	Uneven Tunnel5 (244.3 s)	Waterway Short (472.0 s)	Waterway Long (1616.7 s)	Stair (345.5 s)	Bridge1 (382.5 s)
system	LIO-SAM [3]*	0.23	0.16	-	-	6.30	-
	LVI-SAM [47]*	-	0.30	-	-	-	-
	R3LIVE [48]*	-	11.44	4.80	-	4.54	-
	RELEAD [28]*	0.22	0.17	8.5	67.40	**0.57**	-
odom	FAST-LIO2 [19]*	0.18	0.19	10.27	70.26	4.69	-
	DLIO [49]*	0.23	0.13	2.59	68.04	4.89	-
	GenZ-ICP [24]	-	**0.11**	-	-	-	-
odom, ours	SVGD-ICP	7.06	0.14	1.80	39.28	10.1	-
	SVN-ICP	0.25	0.15	2.97	29.86	7.90	92.23
	SVN-ICP+KF-fix	0.13	0.22	1.82	36.80	2.89	45.08
	SVN-ICP+KF	**0.12**	0.21	**1.28**	**15.58**	2.80	**44.27**

[1] Best results are shown in **bold**.
[2] Results indicated by "*" are reported as presented in [42].
[3] We omitted results where the algorithm failed or produced a APE larger than 100m; denoted as "–".
[4] We selected only a subset of test sequences with the `alpha` sensor setting, based on Table 6 in [42].
[5] In the absence of additional measurements, all algorithms fail in urban tunnel and bridge sequences.

Table III: Normalized Norm Error (NNE) and Kullback–Leibler divergence (KL div.) for translation and rotation against 1000 Monte Carlo ICP samples on the `Long Corridor` sequence [7].

Uncertainty	NNE		KL div.		NNE*		KL div.*	
	trans.	rot.	trans.	rot.	trans.	rot.	trans.	rot.
$\hat{\Sigma}_{\text{Censi}}$ [5]	126.5	565.4	1.2e5	5.8e5	101.2	336.6	1.0e5	2.8e5
$\hat{\Sigma}_{\text{COV-3D}}$ [6]	1.486	2.292	62.21	6.3e3	1.088	1.644	12.68	385.8
$\hat{\Sigma}_{\text{SVGD-ICP}}$	1.288	**1.284**	2.067	272.9	0.997	**0.637**	1.580	79.10
$\hat{\Sigma}_{\text{SVN-ICP}}$	1.281	1.855	**1.996**	93.36	0.995	0.852	**1.532**	41.72
$\hat{\Sigma}_{\text{SVN-ICP-KF-fix}}$	27.56	19.02	1.2e3	1.1e3	15.90	9.004	590.7	465.0
$\hat{\Sigma}_{\text{SVN-ICP-KF}}$	**1.196**	1.559	2.181	**25.37**	**0.939**	0.675	1.713	**10.06**

[1] "*" denotes results with a 90th quantil cut-off, removing unstable MC estimates.
[2] Best results are shown in **bold**.
[3] red indicates notable differences when fusing SVN-ICP with fixed noise parameters.

timation of the proposed SVN-ICP, as exemplified by the challenging corridor scenario in Fig. 1. In this environment, the estimated ICP uncertainties along each dimension consistently align with the environmental structure and the robot's motion, enabling realistic Kalman gain scaling in a simple Kalman filter when the LiDAR degrades. Besides, Fig. 1 b) and c) show that the error samples for each state dimension remain bounded within the 3σ confidence intervals. As a result, both APE and overall robustness are significantly improved without the need for hand-crafted noise models, which aligns with our expectation of uncertainty-aware sensor fusion.

In the LIO setup, IMU noise does not influence SVN-ICP's uncertainty estimation but directly affects the Kalman gain. Thus, setting (roughly) realistic IMU noise values reflecting the robot's motion is crucial for odometry performance.

However, while the proposed SVN-ICP provides consistent one-shot uncertainty estimation under common LiDAR degradations—such as poor initialization or geometry—it is still limited in certain scenarios when the robot's motion becomes unobservable. These include unstructured environments with aggressive motion (e.g., Laurel Caverns) and LiDAR-degraded scenes with moving objects (e.g., Urban Tunnel), leading to severe map corruption. In such cases, although the estimated uncertainty using the proposed method increases consequentially, incorporating additional observations, such as the robot's velocity, along with sophisticated map management becomes essential for achieving high-quality odometry performance.

B. Analysis of Uncertainty Estimation

To intuitively assess uncertainty quality, we adopted the Monte Carlo-based method from [6], using 1000 samples to approximate a pseudo-true ICP distribution[4]. On the `Long Corridor` sequence [7], we compare our proposed uncertainty estimates against classic methods[5] [5, 6] by computing the Kullback-Leibler (KL) divergence and Normalized Norm Error (NNE) [6]. While the KL divergence evaluates the approximated ICP posterior distribution, the NNE measures the estimation error normalized by the predicted uncertainty, offering additional insight into the accuracy and consistency of both the ICP estimates and their associated uncertainties.

Results are shown in Table III. It can be observed that our ICP variants outperform all classical methods [5, 6] on both KL divergence and NNE, indicating that our uncertainty estimates are accurate and consistent throughout the entire test

[4][Online] https://github.com/CAOR-MINES-ParisTech/3d-icp-cov.

[5]Both the MC-based and classical ICP methods are initialized with the ground-truth transformation, following [6]. In contrast, our ICP variants start from inaccurate initial state predictions. The pseudo–true covariance estimated with 1000 MC samples provides only an approximate reference for comparison.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

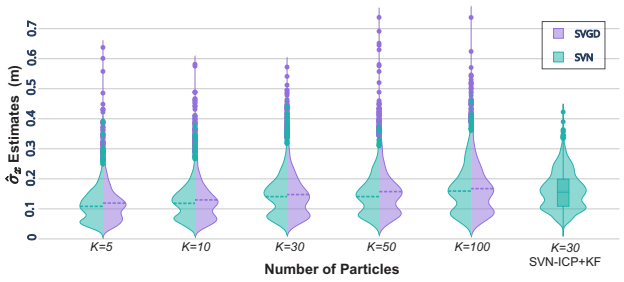

Figure 2: 1σ estimates in the x direction with varying particles K.

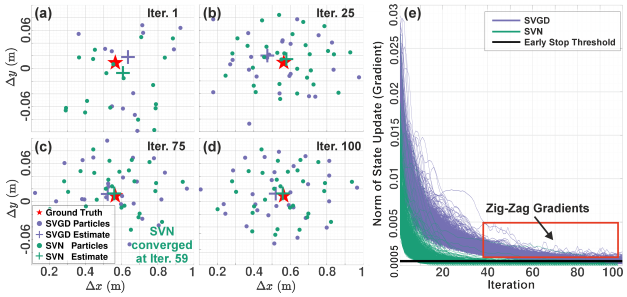

Figure 3: Convergence analysis of SVGD-ICP and SVN-ICP on the `Long Corridor` sequence: (a-d) particle distributions of frame 1136, (e) norms of state updates over successive iterations. While SVN-ICP converges around the 75th iteration, the particles of SVGD-ICP continue to exhibit noticeable movement at 100th iteration.

sequence. Compared to SVGD-ICP, which exhibits slow and unstable convergence, SVN-ICP provides superior uncertainty estimates, achieving lower KL divergence for both translation and rotation components.

Notably, the proposed SVN-ICP in the LIO setting achieves the lowest error metrics, benefiting from both the estimated noise parameters and the accurate initial guess provided by the IMU. In contrast, the significantly higher KL divergence and NNE indicate that a Kalman filter with fixed ICP noise parameters fails to deliver robust and accurate pose estimates.

In addition, Fig. 2 visualizes the 1σ estimates in the x direction obtained using both SVGD-ICP and SVN-ICP with varying particle sizes[6]. It can be observed that the 1σ estimates in the x direction approximate a non-Gaussian noise distribution characterized by multi-modality and long tails, which aligns with the expected LiDAR degradation in a long corridor environment in Fig. 1. Moreover, SVGD-ICP produces larger noise in the x direction due to unstable convergence, while SVN-ICP, fused with a Kalman filter, suppresses long-tail effects thanks to accurate initialization using IMU.

C. Ablation Study on SVN-ICP

1) Convergence Analysis of SVGD and SVN: Fig. 3 shows SVGD and SVN convergence on the `Long Corridor` sequence via particle distributions and state-update norms.

Compared to SVN, which converges rapidly within the first 30 iterations (Fig. 3b), SVGD-ICP shows slower and sometimes ill-conditioned gradient descent, leading to zigzag particle trajectories. As illustrated in Fig. 3a, all SVN-ICP particles stabilize by the 59th iteration, whereas those of SVGD-ICP continue moving until the maximum iteration. This demonstrates the limitation of gradient-based optimization under challenging conditions. Consequently, SVGD-ICP generally requires more iterations and runtime (Table IV) and often fails to meet the early-stopping criterion.

2) Particle Size and Runtime: Table IV compares SVGD-ICP and SVN-ICP with different particle counts to assess the trade-off between efficiency and uncertainty accuracy.

As the number of particles increases, both SVGD and SVN achieve lower KL divergence and NNE in solving ICP, reflecting improved distribution approximation—albeit with higher computational cost, as expected for non-parametric, sampling-based methods. However, this improvement does not translate to better ICP accuracy, which remains comparable for SVGD-ICP and SVN-ICP across all particle counts. Overall, SVN-ICP demonstrates superior performance in terms of computation time and convergence speed.

[6]Only the x-axis is shown due to space constraints, as it is the most affected dimension in the corridor scenario.

Table IV: Ablation study on number of particles K for both SVGD- and SVN-ICP on the `Long Corridor` sequence.

	Metric	Number of Particles					
		1	**5**	**10**	**30**	**50**	**100**
SVGD-ICP	APE (m)	1.165	1.267	1.484	1.324	1.378	1.349
	Runtime (s)	0.486	0.500	0.667	0.611	0.771	1.209
	Avg. iter.	96	98	99	99	99	96
	KL div.	- / -	5.00 / 928.5	1.59 / 209	1.58 / 79.1	1.62 / 48.3	1.66 / 7.28
	NNE	- / -	1.30 / 1.46	1.19 / 1.03	0.99 / 0.64	0.93 / 0.58	0.88 / 0.53
SVN-ICP	APE (m)	1.205	1.256	1.270	1.321	1.268	1.383
	Runtime (s)	0.251	0.337	0.369	0.469	0.519	0.691
	Avg. iter.	33	56	65	65	64	62
	KL div.	- / -	3.94 / 1073	1.58 / 207	1.53 / 41.7	1.58 / 48.3	1.60 / 13.7
	NNE	- / -	1.37 / 1.76	1.21 / 1.22	0.99 / 0.85	0.97 / 0.77	0.88 / 0.70

[1] KL div. and NNE computed with the 90th MC quantil cut-off (see Table III).
[2] KL div. and NNE are reported in translation / rotation format.
[3] Runtime measures the computation time of Algorithm I, excluding all pre- and postprocessing steps.

We used 30 particles in all experiments across different test sequences. However, as shown in Fig. 2, the shape of the approximated distributions remains largely invariant to the number of particles. This suggests that using just 5 to 10 particles is sufficient for robust sensor fusion, while preserving computational efficiency for online applications.

VII. Conclusion

This letter presents SVN-ICP, a novel ICP-based LiDAR odometry method leveraging Stein Variational Newton, enabling accurate transformation and consistent uncertainty estimation. Evaluated on challenging datasets against state-of-the-art methods, the proposed approach achieves superior accuracy and robustness, highlighting that realistic uncertainty estimation is essential for robust sensor fusion.

However, the current implementation of SVN-ICP employs a basic odometry design with a simple Kalman filter and naïve map management, pointing to future research topics toward highly efficient, uncertainty-aware LiDAR odometry that supports active SLAM. In addition, ongoing work also includes investigating the kernel parameterization and extending the SVN-ICP to other range sensors, such as radar.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

ACKNOWLEDGMENT

Special thanks to Shibo Zhao and Honghao Zhu from the AirLab at Carnegie Mellon University for their support with the `SubT-MRS` dataset and insightful discussions on the method. We also thank Fahira Afzal Maken and Prof. Fabio Ramos from the School of Computer Science at the University of Sydney for their valuable input during the early stages of this work.

REFERENCES

[1] D. Lee, M. Jung, W. Yang, and A. Kim, "LiDAR odometry survey: recent advancements and remaining challenges," *Intell. Serv. Robot.*, vol. 17, 2024.

[2] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, 1992.

[3] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020.

[4] Y. Wu, T. Guadagnino, L. Wiesmann, L. Klingbeil, C. Stachniss, *et al.*, "LIO-EKF: High frequency LiDAR-Inertial odometry using extended Kalman filters," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[5] A. Censi, "An accurate closed-form estimate of ICP's covariance," in *2007 IEEE Int. Conf. Robot. Autom. (ICRA)*, 2007.

[6] M. Brossard, S. Bonnabel, and A. Barrau, "A new approach to 3D ICP covariance estimation," *IEEE Robot. and Autom. Lett.*, 2020.

[7] S. Zhao, Y. Gao, T. Wu, S. Singh, Scherer, *et al.*, "SubT-MRS dataset: Pushing SLAM towards all-weather environments," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[8] F. A. Maken, F. Ramos, and L. Ott, "Stein ICP for uncertainty estimation in point cloud matching," *IEEE Robot. and Autom. Lett.*, 2022.

[9] Q. Liu and D. Wang, "Stein variational gradient descent: a general purpose Bayesian inference algorithm," in *2016 Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016.

[10] U. Marteau-Ferey, F. Bach, and A. Rudi, "Globally convergent Newton methods for ill-conditioned generalized self-concordant losses," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019.

[11] G. Detommaso, T. Cui, A. Spantini, Y. Marzouk, and R. Scheichl, "A Stein variational Newton method," in *2018 Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018.

[12] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-LOAM: Fast lidar odometry and mapping," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021.

[13] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot.: Sci. Syst. (R:SS)*, 2018.

[14] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2003.

[15] S. Fontana, D. Cattaneo, A. L. Ballardini, M. Vaghi, and D. G. Sorrenti, "A benchmark for point clouds registration algorithms," *Robot. Auton. Syst.*, vol. 140, 2021.

[16] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: Real-time elastic LiDAR odometry with loop closure," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2022.

[17] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In defense of point-to-point ICP – simple, accurate, and robust registration if done the right way," *IEEE Robot. and Autom. Lett.*, vol. 8, no. 2, 2023.

[18] S. Ferrari, L. D. Giammarino, L. Brizi, and G. Grisetti, "MAD-ICP: It is all about matching data – robust and informed LiDAR odometry," *IEEE Robot. and Autom. Lett.*, vol. 9, no. 11, 2024.

[19] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Trans. on Robot.*, vol. 38, no. 4, 2022.

[20] W. Du and G. Beltrame, "Real-time simultaneous localization and mapping with LiDAR intensity," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023.

[21] P. Pfreundschuh, H. Oleynikova, C. Cadena, R. Siegwart, and O. Andersson, "COIN-LIO: Complementary intensity-augmented LiDAR inertial odometry," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[22] Y. Wu, D. J. Yoon, K. Burnett, S. Kammel, Y. Chen, H. Vhavle, and T. D. Barfoot, "Picking up speed: Continuous-time lidar-only odometry using Doppler velocity measurements," *IEEE Robot. and Autom. Lett.*, vol. 8, no. 1, 2023.

[23] J. Huang, Y. Zhang, Q. Xu, S. Wu, J. Liu, and oters, "LA-LIO: Robust localizability-aware LiDAR-inertial odometry for challenging scenes," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2024.

[24] D. Lee, H. Lim, and S. Han, "GenZ-ICP: Generalizable and degeneracy-robust LiDAR odometry using an adaptive weighting," *IEEE Robot. and Autom. Lett.*, vol. 10, no. 1, 2025.

[25] T. Tuna, J. Nubert, Y. Nava, S. Khattak, and M. Hutter, "X-ICP: Localizability-aware LiDAR registration for robust localization in extreme environments," *IEEE Trans. on Robot.*, vol. 40, 2024.

[26] J. Hatleskog and K. Alexis, "Probabilistic degeneracy detection for point-to-plane error minimization," *IEEE Robot. and Autom. Lett.*, vol. 9, no. 12, 2024.

[27] H. Shen, Z. Wu, Y. Hui, W. Wang, Q. Lyu, T. Deng, Y. Zhu, B. Tian, and D. Wang, "CTE-MLO: Continuous-time and efficient multi-LiDAR odometry with localizability-aware point cloud sampling," *IEEE Trans. Field Robot.*, vol. 2, 2025.

[28] Z. Chen, H. Chen, Y. Qi, S. Zhong, D. Feng, J. Wu, W. Wen, and M. Liu, "RELEAD: Resilient localization with enhanced LiDAR odometry in adverse environments," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[29] M. Nissov, N. Khedekar, and K. Alexis, "Degradation resilient LiDAR-Radar-Inertial odometry," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[30] S. Zhao, H. Zhu, Y. Gao, B. Kim, Y. Qiu, A. M. Johson, and S. Scherer, "SuperLoc: The key to robust LiDAR-Inertial localization lies in predicting alignment risks," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025.

[31] J. Nubert, E. Walther, S. Khattak, and M. Hutter, "Learning-based localizability estimation for robust LiDAR localization," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2022.

[32] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[33] D. Landry, F. Pomerleau, and P. Giguère, "CELLO-3D: Estimating the covariance of ICP in the real world," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.

[34] A. De Maio and S. Lacroix, "Deep Bayesian ICP covariance estimation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2022.

[35] F. A. Maken, F. Ramos, and L. Ott, "Bayesian iterative closest point for mobile robot localization," *Int. J. Robot. Res. (IJRR)*, vol. 41, no. 9-10, pp. 851–874, 2022.

[36] ——, "Speeding up iterative closest point using stochastic gradient descent," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.

[37] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, 1951.

[38] J. L. Blanco-Claraco, "A tutorial on $\mathbf{SE}(3)$ transformation parameterizations and on-manifold optimization," 2022.

[39] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Trans. on Robot.*, vol. 30, no. 3, 2014.

[40] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, 2022.

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[42] Z. Chen, Y. Qi, D. Feng, X. Zhuang, H. Chen, and oters, "Heterogeneous LiDAR dataset for benchmarking robust localization in diverse degenerate scenarios," *Int. J. Robot. Res. (IJRR)*, 2025.

[43] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-Inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. and Autom. Lett.*, vol. 6, no. 2, 2021.

[44] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robot. and Autom. Lett.*, vol. 7, no. 2, 2022.

[45] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Int. Conf. Learn. Represent. (ICLR)*, 2015.

[47] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled Lidar-Visual-Inertial odometry via smoothing and mapping," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021.

[48] J. Lin and F. Zhang, "R3LIVE: A robust, real-time, RGB-colored, Lidar-Inertial-Visual tightly-coupled state estimation and mapping package," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2022.

[49] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct Lidar-Inertial odometry: Lightweight LIO with continuous-time motion correction," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023.