

# Dynamic Initialization for LiDAR-inertial SLAM

Jie Xu\*, Yongxin Ma\*, Yixuan Li, Xuanxuan Zhang, Jun Zhou<sup>†</sup>, Shenghai Yuan<sup>†</sup>, and Lihua Xie, *Fellow, IEEE*

**Abstract**—The accuracy of the initial state, including initial velocity, gravity direction, and IMU biases, is critical for the initialization of LiDAR-inertial SLAM systems. Inaccurate initial values can reduce initialization speed or lead to failure. When the system faces urgent tasks, robust and fast initialization is required while the robot is moving, such as during the swift assessment of rescue environments after natural disasters, bomb disposal, and restarting LiDAR-inertial SLAM in rescue missions. However, existing initialization methods usually require the platform to remain stationary, which is ineffective when the robot is in motion. To address this issue, this paper introduces a robust and fast dynamic initialization method for LiDAR-inertial systems (D-LI-Init). This method iteratively aligns LiDAR-based odometry with IMU measurements to achieve system initialization. To enhance the reliability of the LiDAR odometry module, the LiDAR and gyroscope are tightly integrated within the ESIKF framework. The gyroscope compensates for rotational distortion in the point cloud. Translational distortion compensation occurs during the iterative update phase, resulting in the output of LiDAR-gyroscope odometry. The proposed method can initialize the system no matter the robot is moving or stationary. Experiments on public datasets and real-world environments demonstrate that the D-LI-Init algorithm can effectively serve various platforms, including vehicles, handheld devices, and UAVs. D-LI-Init completes dynamic initialization regardless of specific motion patterns. To benefit the research community, we have open-sourced our code and test datasets on GitHub<sup>1</sup>.

**Index Terms**—Dynamic initialization, LiDAR-inertial, LiDAR-gyroscope odometry, simultaneous location and mapping.

## I. INTRODUCTION

LiDAR-Inertial Odometry (LIO) [1]–[3] is a key technology in modern field robotics, but its reliance on stationary initialization poses challenges in urgent scenarios like restarting LIO in highway autonomous driving conditions [4] or bomb disposal robot rushing to area of interest. The key issue is that the IMU mitigates motion distortion in LiDAR point clouds through high-frequency motion data, improving SLAM accuracy. However, successful initialization depends

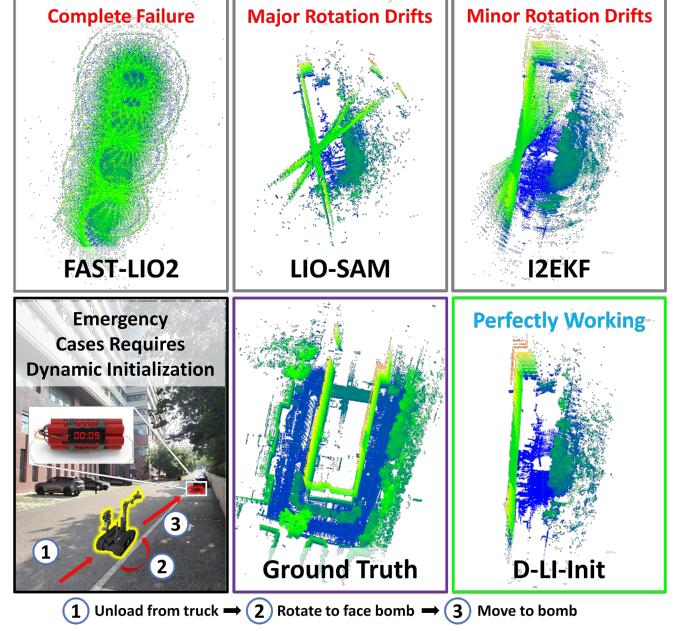


Fig. 1: Demonstration of real-world scenarios where the robot requires dynamic initialization, along with point cloud maps created using SOTA algorithms and ours.

on precise estimates of velocity, gravity direction, and IMU biases. Notably, the gravity direction represents a unit vector indicating the direction of gravity. Inaccurate initial estimates, especially for gravity direction and velocity, prevent the IMU from correcting motion distortion, potentially leading to faulty point cloud registration and system errors, as shown in Fig. 1.

Existing research on LIO initialization [5], [6] is limited, primarily focusing on estimating the initial state by keeping the robot stationary for a period of time. Using those methods while the robot is in motion can result in incorrect initial estimates. This motivates the development of a robust dynamic initialization method for LIO. We hypothesize that without known initial values, the system cannot perform SLAM properly. This requires an analysis of the relationship between LiDAR odometry and IMU measurements to solve for the initial values. However, this approach poses several challenges, including (1) achieving highly accurate LiDAR odometry, (2) managing the data from both LiDAR odometry and IMU measurements to accurately solve for the initial values, and (3) ensuring that the initialization method is not constrained by specific motion patterns.

This paper presents D-LI-Init, a novel method that aligns LiDAR odometry with IMU pre-integration for accurate initial

\* denotes equal contribution, <sup>†</sup> denotes corresponding author.

Jie Xu, Yongxin Ma and Jun Zhou are with school of Mechanical Engineering, Shandong University, Jinan 250061, China and Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Ministry of Education, Jinan 250061, China (e-mail: jeff\_xu\_0503@foxmail.com; yxma@mail.sdu.edu.cn; zhousjun@sdu.edu.cn).

Jie Xu, Shenghai Yuan, and Lihua Xie are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore (e-mail: jeff\_xu\_0503@foxmail.com; shyuan@ntu.edu.sg; elhxie@ntu.edu.sg).

Yixuan Li is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, 710049, China (e-mail: 2191312118@stu.xjtu.edu.cn).

Xuanxuan Zhang is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 430072, China (email: xuanxuanzhang@whu.edu.cn).

<sup>1</sup> <https://github.com/lian-yue0515/D-LI-Init>

state estimation in both stationary and moving robots. Enhancing the robustness and accuracy of LiDAR odometry is crucial for the effectiveness of this method. When the initial state, like gravity direction or velocity, is unknown, accelerometer data becomes unreliable due to interference from gravity. In contrast, the gyroscope is insensitive to initial state uncertainties, allowing its observational data to be correctly employed. As a result, LiDAR-only odometry (LO) is upgraded to LiDAR-gyroscope odometry (LGO) by tightly integrating gyroscope and LiDAR data through the error state iterated Kalman filter (ESIKF) [5], [7], improving overall accuracy and performance. This approach produces highly accurate LiDAR odometry outputs and improves the precision of initial value estimation during the alignment process. Subsequently, the rough initial state is integrated into the LIO system to obtain more precise odometry outputs, which are iteratively refined using IMU measurements, ultimately resulting in a high-precision initial state. Proposed D-LI-Init has been incorporated into FAST-LIO, and experimental results demonstrate its effectiveness in accurately estimating initial values during robot motion, thereby ensuring precise subsequent state estimations (Fig. 1). The main contributions of this paper are as follows:

- We propose a dynamic initialization method for LiDAR-inertial SLAM systems. This method iteratively aligns LiDAR-based odometry with IMU measurements, enabling fast and robust estimation of the system's initial state without requiring specific motion patterns.
- A tightly coupled integration of LiDAR and gyroscope is achieved using the ESIKF framework. The gyroscope compensates for rotational distortion in point clouds, while translational distortion is addressed during the iterative update process, resulting in improved point cloud quality and enhanced precision in state estimation. Experimental results show that this method outperforms LO in robustness and accuracy.
- We evaluate the dynamic initialization performance on public datasets and in real-world environments across various platforms, including vehicles, handheld devices, and UAVs. The experimental results demonstrate that the proposed method excels in dynamic initialization across various scenarios.
- We will open-source the code and test datasets, providing a benchmark for future research in dynamic initialization. This is the first open-source package for dynamic initialization in LiDAR-inertial SLAM systems.

## II. RELATED WORKS

Robust initialization is a critical challenge in SLAM systems [8]–[10], particularly for LiDAR-inertial SLAM, which relies on estimating initial variables through analyzing the relationship between LiDAR odometry and inertial data integration. This section explores existing work on initialization methods and LiDAR odometry.

### A. Initialization Methods

In both LiDAR and visual-inertial SLAM, key initial variables such as velocity, gravity direction, and IMU biases are

equally crucial for accurate system performance. [11]. Existing LiDAR-inertial SLAM studies primarily rely on stationary initialization approaches [5], [12]–[14], where the system must remain static for a period. During this time, gyroscope biases are estimated based on the average angular velocity of the IMU, while the difference between average acceleration and local gravitational acceleration is used to determine the accelerometer bias. The gravity direction is inferred from this vector. While this stationary approach works well for vehicle-mounted systems starting from rest, it fails to accommodate systems that need to initialize while in motion, such as emergency response robots and wearable LiDAR SLAM. This limitation highlights the need for more robust dynamic initialization methods that can handle scenarios where the system is moving.

In visual-inertial SLAM, a monocular camera lacks metric scale, requiring the robot to maintain non-zero acceleration for initialization. This led to the development of dynamic methods [15]–[17], such as a tightly coupled closed-form solution [18] that estimates initial state variables and feature depth using visual features and accelerometer data, though at a high computational cost. ORB-SLAM3 [19] handles IMU initialization as a maximum a posteriori estimation, using a three-step process to estimate initial parameters in dynamic situations without requiring a dedicated initialization phase. He et al. [20] introduces a decoupled visual-inertial odometry initialization method, which separates the estimation of rotation and translation. This method achieves higher efficiency and robustness by solving for initial velocity and gravity direction using linear translational constraints in a globally optimal manner. VINS-Fusion [11], [21] aligns vision-only structure from motion with IMU pre-integration measurements [22] to recover the metric scale, velocity, gravity direction, and gyroscope biases. While effective for visual SLAM, these VIO initialization methods are not directly applicable to LiDAR-Inertial Systems, which do not require scale recovery or motion excitation in all directions. However, they provide insights for developing dynamic initialization in LiDAR-inertial SLAM, where initial state estimation can be achieved by analyzing the relationship between LiDAR odometry and inertial data.

### B. LiDAR Odometry

For the dynamic initialization of LiDAR-inertial SLAM systems, the accuracy of the LiDAR odometry significantly influences the precision of the initial state estimation. LOAM [23] is a classic representation of LiDAR odometry, which achieves improved registration accuracy and algorithm efficiency by extracting line and plane features for registration. However, these methods rely on a constant velocity model for motion distortion compensation, which limits their performance in real-world environments, especially under challenging motion scenarios. CT-ICP [24] introduces the concept of continuous-time sensor motion, modeling it as a continuous function. It determines the vehicle's state at the beginning and end of each LiDAR frame. CT-ICP demonstrates high accuracy but comes with higher computational costs due to the increased dimensionality of the state variables and occasionally has

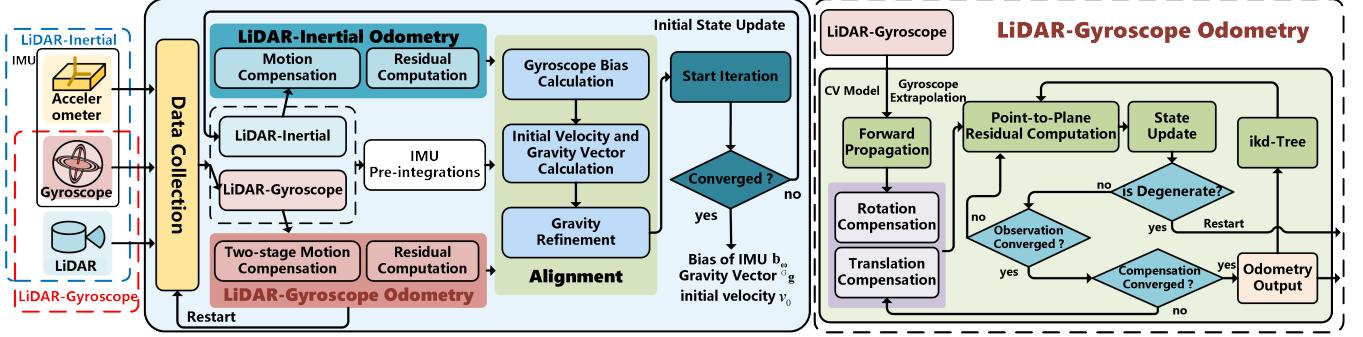


Fig. 2: D-LI-Init overview.

overfitting issues. KISS-ICP [25] provides a lightweight and efficient LiDAR odometry system, which estimates states with point-to-point ICP, ensuring stable operation across different environments and motion patterns. However, its effectiveness drops with sparse point clouds generated from devices like Livox Mid360. I2EKF-LO [26] introduces a dual-iterative extended Kalman filter for odometry estimation. It dynamically adjusts process noise during state estimation and establishes motion models for different sensor platforms, resulting in accurate and efficient state estimation. While existing systems perform reasonably well, their state estimation accuracy is slightly lower than the LiDAR-inertial systems.

For LiDAR-inertial systems [5], [7], which utilizes IMU data to assist in point cloud motion distortion compensation and tightly couples IMU and LiDAR measurements using the ESIKF, high-frequency and high-precision odometry can be output. However, for the LiDAR odometry required in dynamic initialization systems, the initial velocity and gravity direction are unknown, rendering the accelerometer in the IMU unusable. Given that the gyroscope is unaffected by initial values, using ESIKF to achieve tight coupling between the gyroscope and LiDAR could be a novel approach to improving LiDAR odometry accuracy.

### III. METHODOLOGY

In this paper, we present a robust and fast dynamic initialization method for LiDAR-inertial SLAM, dubbed D-LI-Init. The framework, shown in Fig. 2, assumes known extrinsic parameters between sensors, does not necessitate specific motion patterns from the robot and can be adapted to various platforms. To estimate the initial state, an iterative alignment between LiDAR odometry and IMU pre-integration is performed using least squares method (LSM).

#### A. Problem Statement

We define the global frame  ${}^G(\cdot)$  at the initial pose of the LiDAR. The external parameter from LiDAR frame to IMU frame  ${}^I(\cdot)$  is assumed to be known as  ${}^I\mathbf{T}_L = ({}^I\mathbf{R}_L, {}^I\mathbf{t}_L) \in SE(3)$ . Let  ${}^G\mathbf{P}_k({}^G\mathbf{R}_k, {}^G\mathbf{t}_k)$  represent the corresponding pose at the  $k_{th}$  frame, and let the corresponding IMU pre-integration be denoted by  $\alpha_{k+1}^k, \beta_{k+1}^k, \gamma_{k+1}^k$ . The objective of this work is to align the LiDAR odometry poses with the IMU pre-integration by solving for the initial state  $\mathbf{x}_I = (\mathbf{b}_\omega, {}^G\mathbf{v}_0, {}^G\mathbf{g})$

using LSM, where  $\mathbf{b}_\omega$ ,  ${}^G\mathbf{v}_0$ , and  ${}^G\mathbf{g}$  represent the gyroscope bias, initial velocity, and gravity direction, respectively.

#### B. LiDAR-gyroscope Odometry

LGO module uses gyroscope data to predict rotational motion, thereby enabling compensation for rotational motion distortion during scans. A constant velocity model is employed to predict translational motion and translational motion distortion compensation is integrated into the iterative update process to enhance point cloud quality. To address the mismatch between the constant velocity model and actual motion during translational prediction, a frame segmentation of the incoming point cloud is adopted [27], [28]. This approach minimizes errors caused by the constant velocity assumption, ensuring high-precision state estimation comparable to LIO. It improves the accuracy of dynamic initialization by providing better initial values for iterative alignment, reducing the required iterations, and enhancing the correctness of the D-LI-Init.

The discrete model is as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k \boxplus (\Delta t f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)), \quad (1)$$

where  $\Delta t$  is the time interval between two scans,  $\boxplus$  denotes the “plus” on the state manifold. The state vector  $\mathbf{x}$ , input  $\mathbf{u}$ , noise  $\mathbf{w}$ , and discrete state transfer function  $f$  are defined as:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} {}^G\mathbf{R} \\ {}^G\mathbf{t} \\ {}^G\mathbf{v} \\ \mathbf{b}_\omega \end{bmatrix}, \quad \mathbf{u} = [\omega_m], \quad \mathbf{w} = \begin{bmatrix} \mathbf{n}_\omega \\ \mathbf{n}_v \\ \mathbf{n}_{b\omega} \end{bmatrix}, \\ \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) &= \begin{bmatrix} \omega_{m_k} - \mathbf{b}_{\omega_k} - \mathbf{n}_{\omega_k} \\ {}^G\mathbf{v}_k \\ \mathbf{n}_{v_k} \\ \mathbf{n}_{b\omega_k} \end{bmatrix}, \end{aligned} \quad (2)$$

where  ${}^G\mathbf{R} \in SO(3)$ ,  ${}^G\mathbf{t}$  are the attitude and position in the global coordinate system (aligned with the first LiDAR frame), respectively, and  ${}^G\mathbf{v}$  is the velocity in the global coordinate system, which is modeled as a random walk process driven by Gaussian noise as  $\mathbf{n}_v$ .  $\mathbf{b}_\omega$  is the gyroscope biases, which is modeled as a random walk process driven by Gaussian noise as  $\mathbf{n}_{b\omega}$ .  $\omega_m$  is the gyroscope measurement, and  $\mathbf{n}_\omega$  is the white noise measured by the IMU.

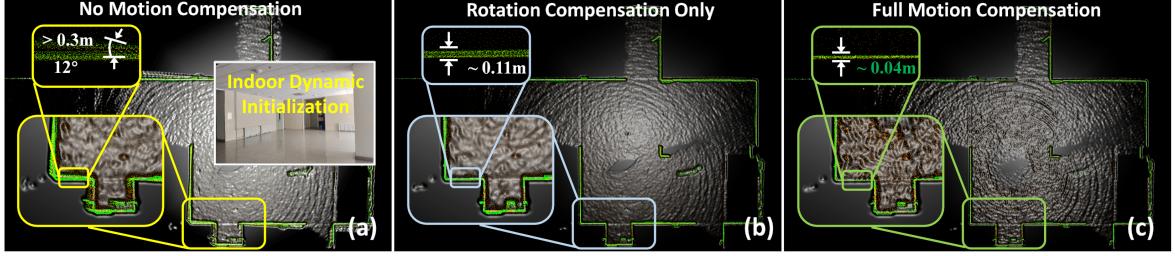


Fig. 3: The effect of the motion distortion compensation method is demonstrated for a simple indoor environment.

The measurement model is directly constructed by calculating point-to-plane distances, eliminating the need for feature extraction from the original point cloud, as follows:

$$0 = \mathbf{h}_j(\mathbf{x}_k, \mathbf{n}_j^L) \triangleq \mathbf{u}_j^T ({}^G\mathbf{R}_k (\mathbf{p}_j^L + \mathbf{n}_j^L) + {}^G\mathbf{t}_k - {}^G\mathbf{q}_j), \quad (3)$$

where  $\mathbf{n}_j^L$  is the measurement noise of LiDAR, and  $\mathbf{p}_j^L$  is the coordinate of a point in the LiDAR frame.  $\mathbf{u}_j^T$  is the normal vector of the plane matching the point in the mapping, and  ${}^G\mathbf{q}_j$  is a point on this plane.

1) *Forward Propagation*: The forward propagation is performed upon receiving gyroscope measurements. More specifically, by setting the process noise to zero, the state and covariance are propagated as follows:

$$\hat{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k \boxplus (\Delta t f(\bar{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{0})), \quad (4)$$

$$\hat{\mathbf{P}}_{k+1} = \mathbf{F}_{\bar{\mathbf{x}}} \bar{\mathbf{P}}_k \mathbf{F}_{\bar{\mathbf{x}}}^T + \mathbf{F}_{\mathbf{w}} \mathbf{Q} \mathbf{F}_{\mathbf{w}}^T, \quad (5)$$

where,  $\bar{\mathbf{x}}$  and  $\bar{\mathbf{P}}$  represent the posterior state and covariance matrix of the frame, respectively, while  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{P}}$  denote the predicted state and covariance matrix,  $\tilde{\mathbf{x}}$  represents the error state. The matrix  $\mathbf{Q}$  is defined as the covariance matrix of  $\mathbf{w}$ . The matrices  $\mathbf{F}_{\bar{\mathbf{x}}}$  are defined as follows:

$$\begin{bmatrix} \text{Exp}(-(\omega_{m_k} - \mathbf{b}_{\omega_k}) \Delta t) & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} \Delta t \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \Delta t & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix},$$

and  $\mathbf{F}_{\mathbf{w}}$  is defined as:

$$\begin{bmatrix} \mathbf{I}_{3 \times 3} \Delta t & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \Delta t & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \Delta t \end{bmatrix}.$$

2) *Backward Propagation and Rotational Distortion Compensation*: The backward propagation process is primarily used to compute the pose of each point within the LiDAR frame. This process focuses solely on the rotational component, denoted as  $\hat{\mathbf{x}}_{k-1} = \hat{\mathbf{x}}_k \boxplus (-\Delta t f(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{0}))$ , which can be simplified accordingly for the purpose of removing rotational distortion. Specifically, this can be expressed as (6).

$$\hat{\mathbf{R}}_{i-1} = {}^{I_k} \hat{\mathbf{R}}_i \text{Exp}\left(\left(\hat{\mathbf{b}}_{\omega_k} - \omega_{m_{i-1}}\right) \Delta t\right) \quad (6)$$

The process determines the relative pose between point  $(i \in (k-1, k])$  within the LiDAR frame and the end of the scan at time  $t_k$ , denoted as  ${}^{I_k} \hat{\mathbf{T}}_i = \left({}^{I_k} \hat{\mathbf{R}}_i, \mathbf{0}\right)$ . This relative

orientation allows for the projection of local measurements  ${}^{L_i} \mathbf{p}_i$  onto the coordinate system  ${}^{L_i} \mathbf{p}_k$  at the end of the scan, thus facilitating the transformation of all points within the LiDAR frame to the coordinate system at the frame's end.

$${}^{L_i} \mathbf{p}_k = ({}^I \mathbf{T}_L^{-1}) \left( {}^{I_k} \hat{\mathbf{T}}_i \right) ({}^I \mathbf{T}_L) {}^{L_i} \mathbf{p}_i \quad (7)$$

where  ${}^I \mathbf{T}_L$  represents a known extrinsic parameter.

3) *Iteration*: Each iteration includes iterative motion distortion compensation and iterative observation update.

i) *Iterative Motion Distortion Compensation*: To correct translational motion distortion in the system, consistent with the backward propagation process, the point cloud of the current frame  $i$  ( $i \in (k-1, k]$ ) is uniformly transformed into the LiDAR coordinate system at the end time  $t_k$ .

For a point  ${}^{L_i} \mathbf{p}_i$  in the LiDAR coordinate system generated at time  $t_i$  within frame  $k$ , given the predicted state transformation  ${}^G \hat{\mathbf{T}}_k$  for frame  $k$  and the posterior state transformation  ${}^G \hat{\mathbf{T}}_{k-1}$  for frame  $k-1$ , the distortion-corrected point  ${}^{L_i} \mathbf{p}_k$  is obtained by determining the transformation  ${}^G \mathbf{T}_i$  of the LiDAR coordinate system in the global coordinate system at time  $t_i$ .

$${}^{L_i} \mathbf{p}_k = ({}^G \hat{\mathbf{T}}_k)^{-1} ({}^G \hat{\mathbf{T}}_i) {}^{L_i} \mathbf{p}_i, \quad (8)$$

$${}^G \hat{\mathbf{T}}_k = [\mathbf{I} \mid {}^G \hat{\mathbf{t}}_i], \quad (9)$$

where  ${}^G \hat{\mathbf{t}}_i$  is predicted by linear interpolation:

$${}^G \hat{\mathbf{t}}_i = {}^G \bar{\mathbf{t}}_{k-1} + \text{scale} ({}^G \mathbf{t}_k - {}^G \mathbf{t}_{k-1}), \quad (10)$$

$$\text{scale} = \frac{t_i - t_{k-1}}{t_k - t_{k-1}}. \quad (11)$$

The effect of the motion distortion compensation method described in this paper is shown in Fig. 3.

ii) *Iterative Observation Update*: The motion-compensated scan, denoted as  $\{{}^{L_i} \mathbf{p}_k\}$ , provides an implicit measurement of the predicted pose  ${}^G \mathbf{T}_{k+1}$ , which is formulated as a point-to-plane distance residual function. Based on this, observation updates are iteratively performed within the ESIKF framework until the estimated state  $\mathbf{x}_{k-1}$  converges. The converged state estimate, denoted as  $\bar{\mathbf{x}}_{k-1}$ , is then used to propagate subsequent IMU measurements. For more details on this iterative observation update process, refer to [5].

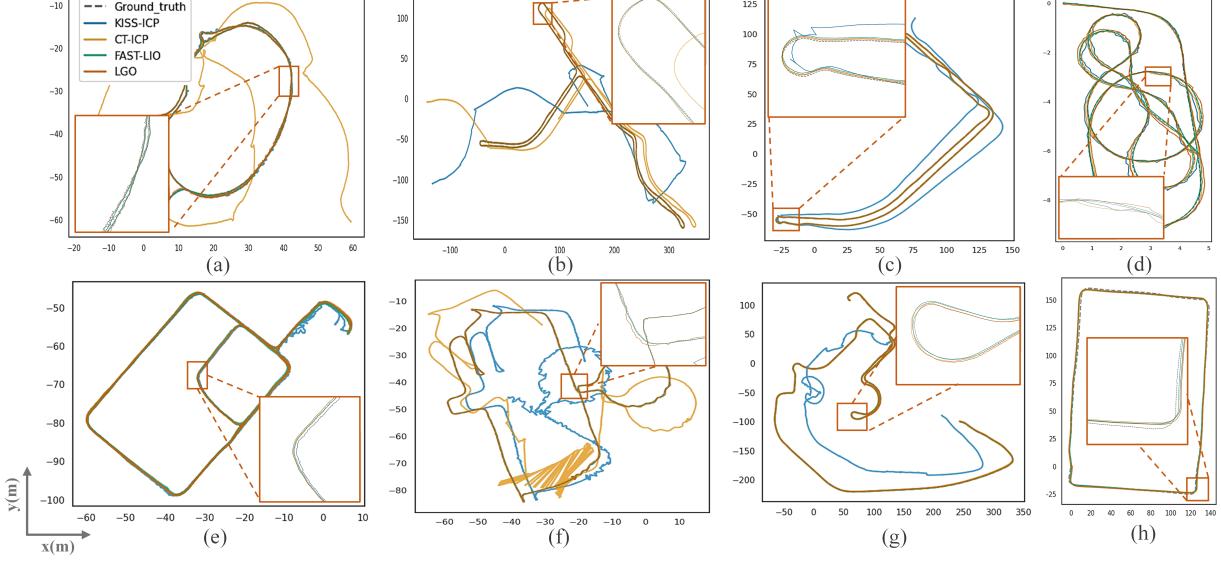


Fig. 4: Trajectory generated by various algorithms for different datasets. LGO performs well, outperforming LO and approaching LIO. The subfigures depict the following: (a) - (h) trajectories of various algorithms in the NEW\_quad, MCD\_night\_04, MCD\_night\_13, Gnd\_MBLR\_1, NEW\_sloitter, NEW\_math, MCD\_day\_02, UbanLoco\_test, respectively.

4) *Degeneracy Detection:* For the LGO module, degeneracy detection ensures that the module provides high-precision LiDAR odometry while also serving as a criterion for determining the reliability of the data. The local submatrix of the covariance matrix better accounts for the rotational and translational constraints [29]. As the rotational and translational matrix blocks clearly contain information about the strength of constraints on the robot's state, this information can be utilized to establish degeneracy detection standards. The covariance matrix can thus be divided into several submatrices:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{rt} \\ \mathbf{P}_{tr} & \mathbf{P}_{tt} \end{bmatrix}_{6 \times 6}, \quad (12)$$

where  $r$  denotes rotation,  $t$  denotes translation,  $\mathbf{P}_{rr}$  contains only information about the rotation variable, and  $\mathbf{P}_{tt}$  contains only information about the translation variable.

The constraints can be distinctly categorized into translation and rotation. Analyzing the covariance matrix as a whole would lead to the coupling of rotation and translation. Since the scales and types of rotation and translation are different, this coupling complicates the setting of threshold parameters for degeneracy detection. Therefore, eigenvalue decomposition is performed separately for rotation and translation:

$$\mathbf{P}_{rr} = \mathbf{V}_r \Sigma_r \mathbf{V}_r^\top, \quad \mathbf{P}_{tt} = \mathbf{V}_t \Sigma_t \mathbf{V}_t^\top, \quad (13)$$

where  $\mathbf{V}_r$  and  $\mathbf{V}_t$  are the eigenvectors in the matrix, and  $\Sigma_r$  and  $\Sigma_t$  are diagonal matrices containing information about the eigenvalues of  $\mathbf{P}_{rr}$  and  $\mathbf{P}_{tt}$ .

The eigenvalues in  $\mathbf{V}_r$  and  $\mathbf{V}_t$  provide a direct measure of the system's observed strength with respect to the rotation and translation constraints. By monitoring the maximum eigenvalues  $\lambda_k^{(\max)r}$  and  $\lambda_k^{(\max)t}$  corresponding to each frame's eigenvalue matrix, we can assess whether the system is experiencing degeneracy. The degeneracy thresholds  $\xi^r$  and  $\xi^t$

are empirically determined for this purpose. It is important to note that the relevant threshold settings are closely tied to the sensor type. Therefore, we establish the thresholds through experiments on datasets using various sensor types to ensure the robustness of the degeneracy detection mechanism.

$$O_k = \begin{cases} 1, & \text{if } \lambda_k^{(\max)r} < \xi^r, \lambda_k^{(\max)t} < \xi^t, \\ 0, & \text{otherwise} \end{cases}, \quad (14)$$

where  $O_k$  indicates whether the  $k_{th}$  data system is degenerate. A value of 1 signifies that no degeneracy has occurred, while a value of 0 indicates system degeneracy. If degeneracy is detected, the data collected at that moment is considered unreliable, implying that the available LiDAR data do not provide accurate odometry information, which may result in initialization failure. Consequently, the previously accumulated frames are discarded, and data collection is restarted.

### C. Dynamic LiDAR-inertial Initialization

The primary objective of the D-LI-Init method is to accurately estimate the initial values, including the initial velocity, gravity direction, and IMU biases. In the dynamic initialization process, the accurate estimation of accelerometer biases requires the platform to undergo at least a 30-degree rotation to decouple them from the gravitational acceleration [11], which limits the general applicability of the method. Therefore, the estimation of accelerometer biases is deliberately omitted, with their determination deferred to subsequent estimation systems. The LGO module tightly couples LiDAR observations with gyroscope data to output high-precision odometry. IMU data is provided, and pre-integration is used to obtain corresponding constraint information. Additionally, the LGO module includes a degeneracy detection mechanism to determine whether the

TABLE I: Comparison of ATE (RMSE) in Meter Across Various **Dynamic** Initialization Datasets

Sequences (Starting Time(s))	Motion Patterns	LiDAR Type	Platform	FAST-LIO2	LIO-SAM	KISS-ICP	CT-ICP	I2EKF	D-LI-Init
Gnd_MBLR_1 (28)	Spinning	Velodyne	Vehicle	2.807	0.046	0.076	<b>0.033</b>	2.864	0.041
Gnd_MBLR_3 (30)	Spinning	Velodyne		3.461	1.009	<u>0.081</u>	0.083	4.252	<b>0.079</b>
M2DGR_street (220)	L-R Turns	Velodyne		4.456	0.113	0.230	<u>0.108</u>	6.875	<b>0.101</b>
MCD_day_02 (112)	Turning	Ouster		3.908	0.063	0.070	<b>0.045</b>	0.114	0.073
MCD_night_04 (45)	Turning	Ouster		5.517	18.753	<u>0.056</u>	0.115	0.142	<b>0.011</b>
NTU_spms_01 (265)	Spinning	Ouster	UAV	8.665	216.439	2.496	- <sup>b</sup>	<b>0.073</b>	0.317
NTU_rtp_03 (235)	Gliding	Ouster		× <sup>a</sup>	230.630	<u>0.974</u>	-	2.154	<b>0.092</b>
NEW_math_0 (111)	S-S Swinging	Ouster	Handheld	×	0.038	0.129	<u>0.095</u>	0.104	<b>0.036</b>
NEW_math_1 (20)	S-S Swinging	Ouster		×	0.130	0.133	<u>0.061</u>	0.363	<b>0.027</b>
NEW_park (112)	Turning	Ouster		1.754	0.065	0.141	<u>0.084</u>	0.074	<b>0.064</b>
R3LIVE_seq_1 (5)	F-B Swinging	AVIA		×	0.023	-	-	<u>0.030</u>	<b>0.016</b>

<sup>a</sup> “×” denotes the result drifted. <sup>b</sup> “-” denotes numerical instability leading to crashing either at Start-up or at degeneracy encounter. Best results are **bolded**, and second-best results are underlined. “L-R Turns” denotes Left and Right Turns, “S-S Swinging” denotes Side-to-Side Swinging, “F-B Swinging” denotes Forward and Backward Swinging, “D-LI-Init” denotes FAST-LIO enhanced by D-LI-Init. The table below follows the same rules.

LiDAR point cloud observations have degenerated. This ensures a reliable evaluation of whether the estimated trajectory is suitable for the dynamic initialization process. Once a sufficient number of trajectory poses are available, the dynamic initialization module is initiated. Initially, a linear system is constructed using LGO data and IMU pre-integration data to solve for the initial parameters, producing a coarse set of initial values. These initial values are then fed back into the LIO system to obtain more precise odometry information, which is subsequently aligned with IMU pre-integration data. Iterative updates are performed, ultimately yielding the initial velocity, gravity direction, and gyroscope biases.

1) *Data Collection*: A sufficient amount of data is initially gathered based on the degeneracy detection criterion. In this study, 20 frames of data are selected to ensure the accuracy of the initial value estimation (This value is determined empirically and can be applied in general scenarios). The corresponding poses are obtained through the LGO module (i.e.,  ${}^G\mathbf{P}_k({}^G\mathbf{R}_k, {}^G\mathbf{t}_k)$ ), and the IMU pre-integration corresponding to the same time intervals (i.e.,  $\alpha_{k+1}^k, \beta_{k+1}^k, \gamma_{k+1}^k$ ) are simultaneously collected for subsequent calculations.

2) *Gyroscope Biases Initialization*: To solve for the gyroscope biases, we align the rotation constraints between two consecutive frames. Given the rotations  ${}^G\mathbf{R}_{k+1}$  and  ${}^G\mathbf{R}_k$  for these frames, and the relative rotation constraints from the IMU pre-integration  $\gamma_{I_{k+1}}^{I_k}$ , we can construct the following linear equation to find the solution:

$$\min_{\delta\mathbf{b}_\omega} \sum_{k=0}^{20} \left\| {}^G\mathbf{R}_{k+1}^{-1} \otimes {}^G\mathbf{R}_k \otimes \gamma_{k+1}^k \right\|^2, \quad (15)$$

$$\gamma_{k+1}^k \approx \hat{\gamma}_{k+1}^k \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{J}_{\mathbf{b}_\omega}^\gamma \delta\mathbf{b}_\omega \end{bmatrix}, \quad (16)$$

where  $\mathbf{J}_{\mathbf{b}_\omega}^\gamma$  denotes the Jacobi matrix of the relative change in rotation with respect to  $\delta\mathbf{b}_\omega$ . The  $\delta\mathbf{b}_\omega$  solved by the above system is equal to the initial  $\mathbf{b}_\omega$ . Subsequently, the initial  $\mathbf{b}_\omega$  is utilized to re-propagate the IMU pre-integration for the solution of the subsequent variables.

3) *Initial Velocity and Gravity Direction Initialization*: The initial velocity and gravity direction are estimated by aligning the velocity and translation constraints derived from

the odometry data and IMU pre-integration. The variables that need to be solved in this process are as follows:

$$\mathbf{x}_{\mathbb{I}} = [{}^G\mathbf{v}_0, {}^G\mathbf{v}_1, \dots, {}^G\mathbf{v}_k, \dots, {}^G\mathbf{v}_{20}, {}^G\mathbf{g}]^T, \quad (17)$$

where  ${}^G\mathbf{v}_k$  is the velocity in LiDAR coordinates at frame  $k$ .  ${}^G\mathbf{g}$  is the gravitational acceleration at the completion of the first LiDAR frame.  $(\cdot)^T$  denotes the transpose of a matrix.

By constructing the state increments in the discrete form of velocity and pose between two frames, the equations for IMU pre-integration [21] are derived as follows:

$$\begin{aligned} \alpha_{k+1}^k &= {}^G\mathbf{R}_k^T ({}^I\mathbf{t}_{k+1} - {}^I\mathbf{t}_k + \frac{1}{2} {}^G\mathbf{g} \Delta t^2 - {}^G\mathbf{v}_k \Delta t), \\ \beta_{k+1}^k &= {}^G\mathbf{R}_k^T ({}^G\mathbf{R}_{k+1} {}^G\mathbf{v}_{k+1} + {}^G\mathbf{g} \Delta t - {}^G\mathbf{R}_k {}^G\mathbf{v}_k) \end{aligned} \quad (18)$$

here,  $\Delta t$  is the time interval between two frames. By combining the LiDAR data with the external parameters of the IMU (i.e.,  ${}^I\mathbf{t}_k = {}^G\mathbf{t}_k - {}^G\mathbf{R}_k {}^I\mathbf{t}_L$ , where  ${}^I\mathbf{t}_L$  denotes the translation external parameter), the following linear model is obtained.

$$\begin{aligned} \hat{\mathbf{z}}_{k+1}^k &= \begin{bmatrix} \hat{\alpha}_{k+1}^k - {}^G\mathbf{R}_k^T ({}^G\Delta\mathbf{t}_{k+1}^k - {}^G\mathbf{R}_{k+1} {}^I\mathbf{t}_L) - {}^I\mathbf{t}_L \\ \hat{\beta}_{k+1}^k \end{bmatrix} \\ &= \mathbf{H}_{k+1}^k \mathbf{x}_{\mathbb{I}_k} + \mathbf{n}_{k+1}^k, \end{aligned} \quad (19)$$

here,  ${}^G\Delta\mathbf{t}_{k+1}^k = {}^G\mathbf{t}_{k+1} - {}^G\mathbf{t}_k$ ,  $\mathbf{x}_{\mathbb{I}_k} = [{}^G\mathbf{v}_k, {}^G\mathbf{v}_{k+1}, {}^G\mathbf{g}]^T$ ,

$$\mathbf{H}_{k+1}^k = \begin{bmatrix} -\mathbf{I} \Delta t & \mathbf{0} & \frac{1}{2} ({}^G\mathbf{R}_k^T) \Delta t^2 \\ -\mathbf{I} & ({}^G\mathbf{R}_k^T) {}^G\mathbf{R}_{k+1} & {}^G\mathbf{R}_k^T \Delta t \end{bmatrix},$$

solve the linear least squares problem to obtain  $\mathbf{x}_{\mathbb{I}}$ :

$$\min_{\mathbf{x}_{\mathbb{I}_k}} \sum_{k=0}^{20} \|\hat{\mathbf{z}}_{k+1}^k - \mathbf{H}_{k+1}^k \mathbf{x}_{\mathbb{I}_k}\|^2. \quad (20)$$

The aforementioned process does not account for the fixed magnitude of gravity, as incorporating this constraint would render the system nonlinear, making it difficult to solve [11]. To address this, the gravity norm is constrained by optimizing the two-dimensional error state in the tangent space of the gravity vector, thereby refining the gravitational acceleration. Finally, all LiDAR frames  ${}^G(\cdot)$  are rotated into the IMU frame  ${}^I(\cdot)$ , enabling seamless integration into the LIO system for subsequent estimation.

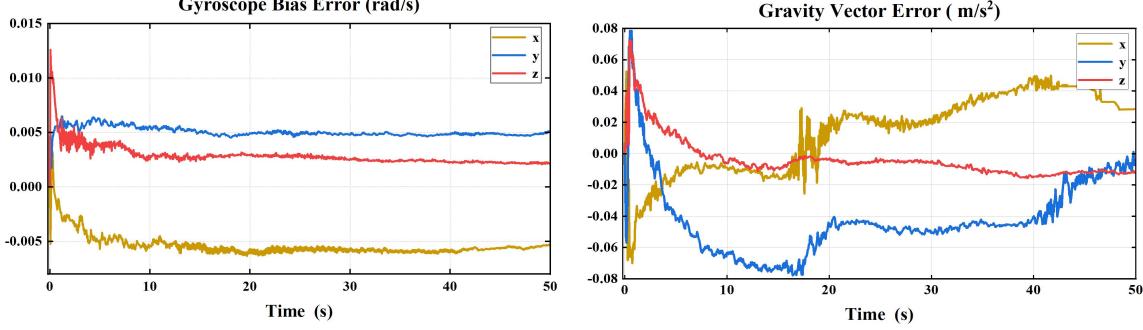


Fig. 5: The error for the D-LI-Init method in estimating the gravity vector and initial gyroscope bias, where the error denotes the difference between the further refined gyroscope bias and gravity vector obtained by the LiDAR-inertial system (initialized using the D-LI-Init method) after completing dynamic initialization and the initial values estimated by the D-LI-Init method.

TABLE II: LGO Performance is Demonstrated through ATE (RMSE) in Meter Comparison.

Sequences	KISS-ICP	CT-ICP	FAST-LIO	LGO
Gnd_MBLR_1	0.103	0.062	<b>0.033</b>	0.061
UrbanLoco_test	2.183	1.297	<u>1.158</u>	<b>1.127</b>
M2DGR_gate	7.063	0.328	<b>0.321</b>	0.334
NEW_sloitter	1.591	0.150	<b>0.119</b>	0.144
NEW_math	9.885	9.811	<b>0.098</b>	0.211
NEW_quad	0.161	20.052	<b>0.066</b>	0.104
MCD_day_02	11.422	0.476	<b>0.466</b>	<b>0.396</b>
MCD_night_04	88.092	1.665	<u>0.862</u>	<b>0.835</b>
MCD_night_13	89.269	55.162	<b>0.873</b>	0.878

TABLE III: Error in Solving for Initial Velocity ( $m/s^2$ )

Sequences	Ground Truth	Solution Results	Error
Gnd_MBLR_1	0.778	0.948	0.170
Gnd_MBLR_3	0.043	0.398	0.355
M2DGR_street	0.772	1.740	0.969
MCD_day_02	1.664	2.663	0.998
MCD_night_04	18.330	19.422	1.092
NEW_math_0	2.035	2.029	0.006
NEW_park	1.364	1.666	0.302

4) *Iterative Update:* To ensure the accuracy of initial value estimation, the primary objective is to achieve higher precision in LiDAR odometry output. In iterative process, the precision of the initial value estimations directly impacts the quality of the point clouds generated by the LiDAR odometry, resulting in a ‘chicken-egg’ problem. Consequently, the coarse initial values obtained through the alignment of LGO and IMU pre-integration are utilized in the LIO system to enhance the precision of the LiDAR odometry. Repeatedly using the above method for the initial values, iterative updates are performed until convergence. For the LIO system, the accuracy of the initial velocity is critical for the quality of pose prediction and point cloud motion distortion compensation based on IMU. Therefore, the convergence of the iteration process is determined by the accuracy of the initial velocity:

$$\left| \left\| G\mathbf{v}_0^{(i)} \right\|^2 - \left\| G\mathbf{v}_0^{(i-1)} \right\|^2 \right| < \Gamma_v, \quad (21)$$

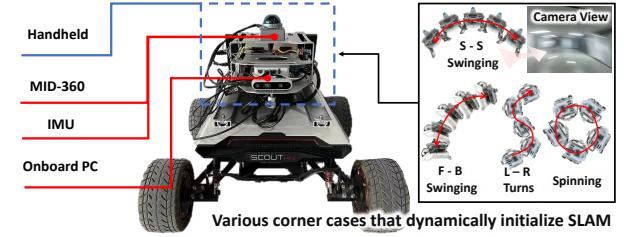


Fig. 6: Vehicle-mounted and Handheld Devices.

where,  $G\mathbf{v}_0^{(i)}$  represents the result of solving the initial velocity for the  $i_{th}$  iteration, and  $\Gamma_v$  denotes the threshold for the difference in the norms between the two initial velocity solutions. This threshold reflects the magnitude of change between the results of successive solutions.

#### IV. EXPERIMENT RESULTS

The performance of the proposed D-LI-Init method is experimentally evaluated on both open-source datasets and in real-world environments. All experiments are conducted on a laptop equipped with an Intel Core i7-9750H (2.60GHz) CPU.

##### A. Public Dataset Validation

We present the performance of the D-LI-Init method and LGO module on open-source datasets. Due to the current scarcity of LiDAR-inertial datasets with dynamic initialization, we evaluated the algorithm by starting SLAM at a specific point during the dataset sequences, simulating the initialization of SLAM while the system is in motion. 16 sequences from notable open-source datasets, including M2DGR [30], MCD [31], UrbanLoco [32], Newer Collegec [33], Ground-challenge [34], and NTU-VIRAL [35], are selected and modified accordingly for this purpose. The chosen sequences encompass a variety of motion platforms, including handheld devices, vehicles, and UAV. They also cover diverse movement patterns such as vehicle rotation, high-speed turns, ascending and descending stairs with handheld devices, and gliding with UAV. These scenarios are selected to demonstrate the reliability and versatility of the proposed algorithm.

TABLE IV: ATE (RMSE) in Meter Comparison of Real-World Proprietary Datasets (Indoor)

Sequences	Motion Patterns	Platform	Initial Velocity	Average Depth	FAST-LIO	LIO-SAM	KISS-ICP	I2EKF	D-LI-Init
Indoor_1	L-R Turns	Vehicle	0.525	4.256	×	0.069	0.058	0.031	<b>0.018</b>
Indoor_2	Spinning		0.693	7.118	×	11.065	0.091	0.273	<b>0.016</b>
Indoor_3	L-R Turns		1.292	7.052	0.087	0.022	0.063	0.026	<b>0.022</b>
Indoor_4	L-R Turns		1.985	6.916	×	0.046	0.064	<b>0.032</b>	0.033
Indoor_5	Spinning		0.797	5.686	5.530	0.025	0.097	0.242	<b>0.019</b>
Indoor_6	Spinning		0.452	7.269	2.548	0.037	0.090	0.080	<b>0.019</b>
Indoor_7	L-R Turns		0.812	4.811	2.679	0.024	0.068	0.120	<b>0.018</b>
Indoor_8	L-R Turns		0.953	6.192	×	0.223	0.109	0.080	<b>0.016</b>
Indoor_9	L-R Turns		0.549	5.595	2.597	0.053	0.142	0.049	<b>0.011</b>
Indoor_10	L-R Turns		1.141	4.196	×	0.127	0.279	0.289	<b>0.027</b>
Indoor_11	Spinning		0.663	6.472	×	0.040	0.063	×	<b>0.018</b>
Indoor_12	L-R Turns		1.208	5.526	×	1.209	0.471	1.200	<b>0.023</b>
Indoor_13	Spinning		0.40	5.68	×	0.034	0.162	0.093	<b>0.019</b>
Indoor_14	L-R Turns		0.609	7.266	4.981	0.041	0.058	0.027	<b>0.023</b>
Indoor_15	F-B Swinging	Handheld	2.135	4.311	×	0.517	0.209	0.235	<b>0.019</b>
Indoor_16	S-S Swinging		1.592	4.156	0.151	0.237	0.191	0.027	<b>0.022</b>
Indoor_17	F-B Swinging		1.887	4.363	×	0.126	0.163	0.038	<b>0.028</b>
Indoor_18	S-S Swinging		1.034	5.328	×	0.055	0.155	0.210	<b>0.024</b>
Indoor_19	S-S Swinging		1.896	3.367	×	0.153	0.394	1.379	<b>0.037</b>
Indoor_20	F-B Swinging		1.152	5.273	×	0.032	0.113	0.043	<b>0.017</b>
Indoor_21	S-S Swinging		1.657	4.890	×	0.054	0.181	1.147	<b>0.027</b>
Indoor_22	Turning		1.966	3.928	×	0.192	0.134	<b>0.026</b>	0.028
Indoor_23	Turning		0.802	5.266	×	0.045	0.058	0.029	<b>0.020</b>
Indoor_24	S-S Swinging		0.844	4.629	2.322	0.100	0.252	0.590	<b>0.027</b>
Indoor_25	S-S Swinging		0.676	4.083	×	0.055	0.113	0.035	<b>0.034</b>
Indoor_26	F-B Swinging		1.694	6.120	7.098	0.368	0.103	0.034	<b>0.024</b>
Indoor_27	Turning		2.386	7.045	×	0.067	0.169	0.451	<b>0.029</b>
Indoor_28	F-B Swinging		2.179	6.912	×	0.050	0.088	8.192	<b>0.022</b>

Average depth refers to the mean distance of all points in a frame. The units of Initial Velocity and Average Depth are  $m/s$  and  $m$ , respectively.

TABLE V: ATE (RMSE) in Meter Comparison of Real-World Proprietary Datasets (Outdoor)

Sequences	Motion Patterns	Platform	Initial Velocity	Average Depth	FAST-LIO	LIO-SAM	KISS-ICP	I2EKF	D-LI-Init
Outdoor_1	L-R Turns	Vehicle	1.214	12.956	×	1.918	0.101	2.933	<b>0.012</b>
Outdoor_2	L-R Turns		1.150	13.363	×	0.022	0.040	0.020	<b>0.012</b>
Outdoor_3	Spinning		1.632	14.087	×	1.686	0.049	0.056	<b>0.016</b>
Outdoor_4	Spinning		0.667	6.927	9.525	0.041	0.099	2.633	<b>0.016</b>
Outdoor_5	Spinning		0.809	20.941	0.087	0.028	0.086	0.029	<b>0.018</b>
Outdoor_6	L-R Turns		1.079	15.486	×	4.143	0.099	0.165	<b>0.041</b>
Outdoor_7	L-R Turns		1.095	19.901	×	0.027	0.069	0.038	<b>0.018</b>
Outdoor_8	Spinning		0.807	20.840	0.044	0.019	0.047	0.021	<b>0.017</b>
Outdoor_9	L-R Turns		1.500	15.757	×	0.025	0.068	0.395	<b>0.015</b>
Outdoor_10	L-R Turns		1.708	14.256	×	0.697	0.074	0.102	<b>0.019</b>
Outdoor_11	Spinning		2.223	13.339	×	0.062	0.067	0.041	<b>0.024</b>
Outdoor_12	L-R Turns		1.405	11.399	×	0.659	0.062	0.041	<b>0.025</b>
Outdoor_13	L-R Turns		1.155	14.871	×	0.034	0.084	3.395	<b>0.016</b>
Outdoor_14	Spinning		0.780	19.474	×	0.027	0.121	0.161	<b>0.014</b>
Outdoor_15	L-R Turns		1.280	18.289	×	5.134	0.207	×	<b>0.020</b>
Outdoor_16	L-R Turns		2.461	6.636	×	0.054	0.292	0.049	<b>0.030</b>
Outdoor_17	L-R Turns		1.041	16.174	×	0.686	0.059	0.922	<b>0.018</b>
Outdoor_18	L-R Turns		0.965	14.128	×	0.269	0.091	1.311	<b>0.017</b>
Outdoor_19	L-R Turns		2.669	13.708	1.856	0.044	0.058	0.031	<b>0.020</b>
Outdoor_20	Spinning		2.173	12.149	3.447	0.090	0.127	0.097	<b>0.090</b>
Outdoor_21	Spinning		0.891	12.672	×	0.455	0.085	0.299	<b>0.021</b>

1) *Overall Performance:* The higher the accuracy of the initial state estimation, the faster and more precise the subsequent state estimation in LiDAR-inertial SLAM becomes. To quantitatively evaluate the quality of the initial state estimation, we integrated the D-LI-Init method into the FAST-LIO system for state estimation. We collected trajectory data for 10 seconds after initiating SLAM at a specific time period in the dataset (simulating the start of SLAM during motion) and analyzed the absolute trajectory error (ATE) [36] between the generated trajectories and the ground truth.

To highlight the accuracy and robustness of our method,

we compared the results with those from FAST-LIO, LIO-SAM, KISS-ICP, CT-ICP, and I2EKF algorithms, as shown in Table I. FAST-LIO represents tightly-coupled systems in a filter framework, while LIO-SAM serves as an example of a loosely-coupled system in an optimization framework. KISS-ICP, CT-ICP, and I2EKF are included as leading LiDAR-only odometry methods for comparison, evaluating the performance of the LiDAR-inertial SLAM systems enhanced by D-LI-Init.

Experimental results demonstrate that tightly-coupled LiDAR-inertial SLAM systems like FAST-LIO have high demands for accurate initial states, and static initialization

TABLE VI: The Performance of **Static** Initialization is Demonstrated through ATE (RMSE) in Meter Comparison.

Sequences	Motion Patterns	Platform	Average Depth	FAST-LIO	LIO-SAM	KISS-ICP	I2EKF	D-LI-Init
Outdoor_22	Vehicle	Static	13.248	0.00756	0.03326	0.03918	0.02117	<b>0.00645</b>
Outdoor_23			15.557	<b>0.00789</b>	0.02622	0.03918	0.01940	<u>0.01284</u>
Outdoor_24			10.934	<u>0.00961</u>	0.03991	0.04774	0.01447	<b>0.00809</b>
Outdoor_25			19.027	<u>0.01040</u>	0.03235	0.04378	0.02070	<b>0.01034</b>
Indoor_29			5.398	<u>0.00612</u>	0.02422	0.03937	0.01602	<b>0.00505</b>
Indoor_30	Handheld	Static	4.816	<b>0.00777</b>	0.04012	0.16229	0.02582	<u>0.01059</u>
Indoor_31			6.681	<b>0.00690</b>	0.05171	0.09803	0.01958	<u>0.01681</u>
Indoor_32			6.578	<b>0.00592</b>	0.02035	0.03821	0.02271	<u>0.00608</u>
Indoor_33			7.278	<u>0.00475</u>	0.05049	0.03927	0.02427	<b>0.00309</b>

methods often fail to perform accurately or even lead to failure when SLAM is started during motion. In contrast, the LiDAR-inertial SLAM systems, enhanced with D-LI-Init, successfully perform dynamic initialization, ensuring accurate state estimation even when SLAM is initiated during motion. Moreover, the system shows high versatility, being applicable to various motion platforms and types. While the trajectory accuracy of the D-LI-Init enhanced system occasionally slightly lags behind that of CT-ICP, it generally outperforms other methods, ensuring reliable initialization in dynamic scenarios.

2) *LGO Performance:* The performance of the LGO module further affects the accuracy of the initial value estimation. To evaluate LGO module, nine sequences are selected without any special preprocessing, and the ATE between the trajectory of algorithm's output and the ground truth is calculated. To emphasize the accuracy of the LGO module, the results are compared with those of the FAST-LIO, KISS-ICP, and CT-ICP algorithms, as shown in Table II. The corresponding trajectory are shown in Fig. 4. The experimental results demonstrate that the overall performance of LGO surpasses that of LO and is comparable to, or in some cases, even better than LIO.

3) *Initial Values Recovery:* To evaluate the accuracy of the initial velocity estimation, the ground truth velocity is not available in open-source datasets, making direct comparison impossible. Considering that FAST-LIO provides accurate state estimates when initialized from a stationary state, we chose the starting point of the open-source dataset sequence (where the device was stationary) to perform state estimation using FAST-LIO. The estimated velocity at a specific time point, simulating the dynamic initialization of SLAM, is then used as the ground truth for initial velocity. Given that the true velocity and the estimated initial velocity may differ in direction due to being referenced in different coordinate systems, the magnitude of both velocities is compared as the standard for evaluating the initial velocity. The specific results are presented in Table III. The results indicate that the maximum initial velocity error is  $1 \text{ m/s}^2$ , with the overall error being relatively small. This demonstrates that the calculated initial velocity is highly accurate and can serve as a high-quality initial value for subsequent LIO.

To assess the effectiveness of our method in estimating the gravity vector and initial gyroscope biases, we collected further refined gyroscope bias and gravity vector data from LiDAR-inertial SLAM systems initialized using D-LI-Init. The accuracy of our method is indirectly assessed by comparing the differences between these refined values and the initial values

provided by D-LI-Init. As shown in Fig. 5, the subsequent refinement of the initial gyroscope biases and gravity vector is minimal, indicating that the initial values calculated by our method are highly accurate.

### B. Real-world Testing

Ensuring the reliability of the algorithm in real-world scenarios is crucial. Data collection is conducted using handheld and vehicle-mounted devices in both indoor and outdoor environments, with data gathering initiated during device motion and continuing for 10 seconds, as shown in Fig. 6. It is important to emphasize that during this process, high-quality undistorted point clouds are matched against prior maps to obtain the global pose of the current frame, which served as the ground truth trajectory. This trajectory is used to accurately evaluate the performance of the D-LI-Init method. Additionally, the corresponding experimental dataset will be made open-source, marking the first open-source dataset specifically designed to evaluate the initialization capabilities of LIO.

1) *Indoor Environments:* The indoor environment used for testing covers an area of over 1500 square meters on the first floor of an experimental building. The specific results are shown in Table IV. It is demonstrated that the original static initialization module of FAST-LIO is inadequate for dynamic initialization, often leading to system failure and preventing successful startup. In rare cases where initialization is completed, subsequent state estimation exhibited significant errors due to inaccurate initial estimates. In contrast, the D-LI-Init method is able to reliably estimate initial values under various dynamic conditions, ensuring that the system maintained high accuracy in state estimation. Overall, as the initial velocity increases, dynamic initialization becomes more challenging, leading to a slight decrease in accuracy.

2) *Outdoor Environments:* The effectiveness of the proposed D-LI-Init method is also evaluated in an outdoor environment. The data collection vehicle, as shown in Fig. 6, is used to gather data around a building on a university campus. The specific results are presented in Table V. These results demonstrate that the inertial system, enhanced by the D-LI-Init method, can initiate SLAM under various outdoor motion conditions with accurate initial values, ensuring that the system maintains an accuracy within 0.09 meters throughout subsequent operations. By analyzing the data from Tables IV and V it can be observed that as the average depth of the LiDAR-perceived scene increases, the system's robustness improves, and accuracy slightly increases.

TABLE VII: The Time Consumed by Each Module (ms)

Sequences	LGO	LIO	Alignment	Iter.	Ave.
Indoor_1	33.42	41.94	0.0079	2	109.23
Indoor_4	27.18	36.19	0.0068	2	99.52
Indoor_15	30.15	41.12	0.0072	1	70.45
Outdoor_1	32.53	40.04	0.0073	3	143.26

“Iter.” denotes Iteration number, “Ave.” denotes average time per frame. The LGO and LIO modules measure the average time per frame, while the alignment module measures the average time per alignment.

3) *Stationary State Initialization Performance:* Simultaneously, a comparison is conducted between the D-LI-Init method and the stationary initialization method under static conditions, as presented in Table VI. The results of both methods are found to be very close, indicating that the D-LI-Init method is also capable of accurately solving for initial values under stationary conditions, demonstrating its robustness irrespective of the specific motion pattern.

### C. Time Consumption Evaluation

To evaluate the time efficiency of the proposed D-LI-Init method, a series of timing tests are conducted on randomly selected sequences, as shown in Table VII. The results indicate that the majority of the processing time is concentrated in the LiDAR odometry module, while the time consumed by the alignment module is negligible. Therefore, the real-time performance of the LGO module is of paramount importance. The single-frame processing time for the LiDAR odometry module (LGO and LIO) can be maintained at approximately 50ms, which is within an acceptable range. However, the motion state at the initial moment influences the number of iterations required by the D-LI-Init method, potentially increasing the processing time. Considering the total time required for odometry calculation using the 20 frames of data collected during the initialization process is typically around 1.5 seconds, a maximum iteration limit of three is set to ensure that the method’s processing time remains within 3 seconds. This constraint is intended to keep the maximum processing time close to the actual data collection time, thereby ensuring the real-time performance of the initialization module.

## V. CONCLUSION

In this paper, we propose a dynamic initialization method for LiDAR-inertial systems. It has been demonstrated to be applicable across various platforms, providing accurate initial values without being constrained by specific motion patterns. Notably, this method is highly dependent on the accuracy of the LiDAR odometry, and we will conduct further research in the area of enhancing LiDAR odometry accuracy.

## REFERENCES

- [1] J. Liu, Y. Zhang, X. Zhao, Z. He, W. Liu, and X. Lv, “Fast and robust lidar-inertial odometry by tightly-coupled iterated kalman smoother and robocentric voxels,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2024.
- [2] X. Ji, S. Yuan, P. Yin, and L. Xie, “Lio-gvm: an accurate, tightly-coupled lidar-inertial odometry with gaussian voxel map,” *IEEE Robotics and Automation Letters*, 2024.
- [3] K. Huang, J. Zhao, J. Lin, Z. Zhu, S. Song, C. Ye, and T. Feng, “Loglio2: A lidar-inertial odometry with efficient uncertainty analysis,” *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8226–8233, 2024.
- [4] Y. Ma, J. Xu, S. Yuan, T. Zhi, W. Yu, J. Zhou, and L. Xie, “Mm-lins: a multi-map lidar-inertial system for over-degenerate environments,” *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2024.
- [5] W. Xu and F. Zhang, “Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.
- [6] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, “Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.
- [7] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “Fast-lio2: Fast direct lidar-inertial odometry,” *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [8] W. Yang, Y. Zhuang, D. Luo, W. Wang, and H. Zhang, “Vi-hso: Hybrid sparse monocular visual-inertial odometry,” *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6283–6290, 2023.
- [9] T.-M. Nguyen, M. Cao, S. Yuan, Y. Lyu, T. H. Nguyen, and L. Xie, “Viral-fusion: A visual-inertial-ranging-lidar sensor fusion approach,” *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 958–977, 2021.
- [10] X. Wang, X. Li, H. Yu, H. Chang, Y. Zhou, and S. Li, “Givl-slam: A robust and high-precision slam system by tightly coupled gnss rtk, inertial, vision, and lidar,” *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2024.
- [11] T. Qin and S. Shen, “Robust initialization of monocular visual-inertial estimation on aerial robots,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4225–4232.
- [12] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.
- [13] Z. Chen, Y. Xu, S. Yuan, and L. Xie, “ig-lio: An incremental gicp-based tightly-coupled lidar-inertial odometry,” *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1883–1890, 2024.
- [14] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, “Lins: A lidar-inertial state estimator for robust and efficient navigation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8899–8906.
- [15] H. Huang, L. Li, C. Hui, and S.-K. Yeung, “Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [16] C. Campos, J. M. Montiel, and J. D. Tardós, “Inertial-only optimization for visual-inertial initialization,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 51–57.
- [17] J. Domänguez-Conti, J. Yin, Y. Alami, and J. Civera, “Visual-inertial slam initialization: A general linear formulation and a gravity-observing non-linear optimization,” in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2018, pp. 37–45.
- [18] A. Martinelli, “Closed-form solution of visual-inertial structure from motion,” *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.
- [19] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [20] Y. He, B. Xu, Z. Ouyang, and H. Li, “A rotation-translation-decoupled solution for robust and efficient visual-inertial initialization,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 739–748.
- [21] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [22] T. Lupton and S. Sukkarieh, “Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2012.
- [23] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time,” in *Robotics: Science and Systems*, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:18612391>
- [24] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, “Ct-icp: Real-time elastic lidar odometry with loop closure,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5580–5586.
- [25] 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 Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.
- [26] W. Yu, J. Xu, C. Zhao, L. Zhao, T.-M. Nguyen, S. Yuan, M. Bai, and L. Xie, “I2ekf-lo: A dual-iteration extended kalman filter based lidar odometry,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.02190>
- [27] Z. Yuan, F. Lang, T. Xu, and X. Yang, “Sr-lio: Lidar-inertial odometry with sweep reconstruction,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 7862–7869.
- [28] F. Zhu, Y. Ren, and F. Zhang, “Robust real-time lidar-inertial initialization,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3948–3955.
- [29] J. Xu, G. Huang, W. Yu, X. Zhang, L. Zhao, R. Li, S. Yuan, and L. Xie, “Selective kalman filter: When and how to fuse multi-sensor information to overcome degeneracy in slam,” 2024. [Online]. Available: <https://arxiv.org/abs/2412.17235>
- [30] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, “M2DGR: A multi-sensor and multi-scenario slam dataset for ground robots,” *IEEE Robotics and Automation Letters*, pp. 1–1, 2021.
- [31] T.-M. Nguyen, S. Yuan, T. H. Nguyen, P. Yin, H. Cao, L. Xie, M. Wozniak, P. Jensfelt, M. Thiel, J. Ziegenbein, and N. Blunder, “Mcd: Diverse large-scale multi-campus dataset for robot perception,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6 2024. [Online]. Available: <https://mcdviral.github.io/>
- [32] W. Wen, Y. Zhou, G. Zhang, S. Fahandezh-Saadi, X. Bai, W. Zhan, M. Tomizuka, and L.-T. Hsu, “Urbanloco: A full sensor suite dataset for mapping and localization in urban scenes,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2310–2316.
- [33] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, “The newer college dataset: Handheld lidar, inertial and vision with ground truth,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4353–4360.
- [34] J. Yin, H. Yin, C. Liang, H. Jiang, and Z. Zhang, “Ground-challenge: A multi-sensor slam dataset focusing on corner cases for ground robots,” in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2023, pp. 1–5.
- [35] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, “Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint,” *The International Journal of Robotics Research*, vol. 41, no. 3, pp. 270–280, 2022.
- [36] M. Grupp, “evo: Python package for the evaluation of odometry and slam.” <https://github.com/MichaelGrupp/evo>, 2017.