

H-GLIVO: A Hierarchically Tightly-Coupled SLAM Framework for GNSS, LiDAR, Inertial, and Visual Odometry

1st Huirong Cai
School of Electronic Engineering
Beijing University
of Posts and Telecommunications
Beijing, China
caihuirong@bupt.edu.cn

2nd Siyuan Jin
School of Electronic Engineering
Beijing University
of Posts and Telecommunications
Beijing, China
jinsiyuan@bupt.edu.cn

3rd Boyang Lou
School of Electronic Engineering
Beijing University
of Posts and Telecommunications
Beijing, China
luobodan@bupt.edu.cn

4th Enwen Hu*
School of Electronic Engineering
Beijing University
of Posts and Telecommunications
Beijing, China
owen.hu@bupt.edu.cn

Abstract—We present H-GLIVO, a hierarchical GNSS-LiDAR-Inertial-Visual SLAM framework built upon FAST-LIVO2 to address accumulated drift in long-distance operations. Our core contribution is a bidirectional tightly-coupled architecture: the front end directly injects spatiotemporally aligned, robustly gated GNSS measurements into the IESKF update for low-latency drift suppression; concurrently, the back end employs a short sliding-window factor graph to jointly optimize multi-sensor constraints, ensuring local consistency and bounded computational cost via marginalization. Crucially, we propose a back-end-guided correction mechanism that asynchronously reconstructs the front-end voxel map and rectifies the state using the globally optimized trajectory, fundamentally eliminating accumulated error even during GNSS outages. Extensive evaluations on challenging large-scale urban and tunnel datasets demonstrate that H-GLIVO yields substantial improvements in localization accuracy and mapping consistency over established baselines.

Index Terms—SLAM, Multi-Sensor Fusion, Factor Graph Optimization, Backend-Guided Correction, Robust GNSS Fusion

I. INTRODUCTION

Long-term autonomy in complex environments such as urban canyons and transport networks demands three core SLAM properties: high-rate local state estimation, unbiased alignment to a global geographic frame, and sustained trajectory-map consistency.

Modern front ends are predominantly direct-method LIO/LIVO systems. FAST-LIO2 couples LiDAR point-to-plane residuals with IMU preintegration in an error-state Kalman filter, achieving efficient frame-to-map registration on

high-frequency point clouds [1]. FAST-LIVO2 augments this by fusing photometric patch residuals into a unified voxel map via sequential updates [2]. Feature-based alternatives like LVI-SAM jointly optimize sparse features with IMU data [3], while the R3LIVE family integrates color with geometry to enhance map fidelity [4].

Because odometry integrates relative measurements, its error grows with distance. A front end with occasional loop closures cannot maintain absolute alignment. GNSS offers absolute anchors, yet naive fusion of receiver solutions often introduces noise and discontinuities, degrading the high-fidelity relative motion estimate.

Factor-graph frameworks such as iSAM2, combined with on-manifold preintegration, provide a principled basis for scalable, multi-sensor optimization [5]–[7]. The challenge thus shifts from *whether* to fuse GNSS to *how*. The crucial task is to inject absolute GNSS information into the high-rate front end with precise timing, statistically sound weighting, and strict frame consistency, all without compromising numerical stability or map continuity [8].

GNSS-SLAM integration follows three main paths. State-level loose coupling merges receiver poses with LIO states, but struggles with poor satellite visibility [8]. Raw-measurement tight coupling fuses pseudorange and carrier phase data, offering robustness but demanding complex signal processing and initialization [9], [10]. Factor-graph joint optimization, which converts RTK/PPP solutions into probabilistic constraints, offers a practical balance [11]–[13]. Success, however, hinges on overcoming critical implementation gaps: information latency from back-end to front-end, robust gating of outlier measurements, and consistent management of coordinate frames and

Supported by State Key Laboratory of Heavy-duty and Express High-power Electric Locomotive Open Fund Project (No. QZKFKT2024-011).

*Corresponding author.

their uncertainties [2], [8], [9], [11], [13].

To address these challenges, we propose H-GLIVO, a Hierarchically Tightly-Coupled SLAM system built upon FAST-LIVO2 [2]. We overcome the limitations of single-stage fusion through a bidirectional architecture integrating multi-level GNSS fusion. First, the front-end IESKF directly ingests GNSS observations filtered by Mahalanobis distance gating, providing low-latency drift suppression and robust outlier rejection. Concurrently, the back end maintains a Short Sliding-Window Factor Graph that jointly optimizes multi-source constraints; by marginalizing old states, this structure ensures computational efficiency and preserves local trajectory consistency, preventing local deformations caused by rigid global constraints. Finally, to close the loop, we employ a Back-End-Guided Front-End Correction mechanism. This feeds the globally optimized trajectory back to the front-end to asynchronously reconstruct the local voxel map and correct the system state, fundamentally eliminating accumulated errors and map ghosting during long-duration operations.

In summary, our main contributions are as follows:

- 1) **Hierarchical Tightly-Coupled Architecture Based on FAST-LIVO2:** We propose a novel strategy that simultaneously integrates GNSS into the front-end IESKF and the back-end short sliding-window factor graph, balancing real-time drift suppression with high-precision global smoothing.
- 2) **Short Sliding-Window Factor Graph Optimization:** We introduce a short sliding-window back end that fuses global GNSS constraints while using marginalization to bound computational cost and maintain local consistency.
- 3) **Back-End-Guided Active Correction:** We design a mechanism that continuously refines the front-end voxel map using back-end optimization results, actively suppressing map deformation to ensure long-term mapping quality.
- 4) **Extensive Validation and Open Source:** Experiments on large-scale datasets (tunnels, urban canyons) demonstrate superior accuracy and robustness under GNSS-degraded conditions. We will open-source the code to facilitate community development.

II. RELATED WORKS

We review three intertwined topics: high-frequency front-end odometry; drift mitigation via loop closure and GNSS integration; and the front-back information exchange that reveals key architectural gaps.

A. High-Frequency Front-End Odometry

Front ends estimate relative motion using direct or feature-based methods. Direct methods like FAST-LIO2 and FAST-LIVO2 couple LiDAR geometric and visual photometric residuals with IMU data in an IESKF, registering raw sensor data against an incremental voxel map [1], [2]. Feature-based methods like LVI-SAM and R3LIVE jointly optimize sparse geometric and visual features with IMU measurements [3],

[4]. Both paradigms temporally integrate relative observations, causing inevitable error accumulation. Designed for local continuity, they lack an intrinsic absolute alignment mechanism and depend on external modules to correct global drift.

B. Drift Mitigation via Loop Closing

Loop closing imposes long-range relative constraints to correct drift and preserve map topology. Global descriptors like Scan Context enable efficient, rotation-invariant retrieval [14]. Local descriptors like STD and BTC use viewpoint-invariant geometric primitives to achieve high recall under sparse overlap [15], [16]. This approach is limited as its constraints are relative, its updates are asynchronous with latency that grows with map size, and it requires physical revisits, which are not guaranteed in all scenarios.

C. Drift Mitigation via GNSS Integration

GNSS provides absolute position anchors. Architectures range from state-level loose coupling to raw-measurement tight coupling and factor-graph joint optimization [8]–[13]. The ultimate performance hinges on two implementation details often relegated to the back end. First is observation quality control, requiring temporal checks, kinematic prediction, and Mahalanobis gating with robust losses to reject outliers [9], [13], [17]. Second is frame consistency, demanding precise LLA-to-ENU conversion, ENU-odometry alignment, and covariance rotation [10]. Placing these processes in the back end creates a temporal gap, preventing immediate correction in the subsequent frame-to-map registration.

D. Front-End and Back-End Interaction

Modern SLAM architectures separate a high-rate front end from a factor-graph back end that leverages tools like iSAM2 and on-manifold preintegration [5], [7], [8]. A structural limitation is the information flow's latency and asymmetry. Back-end corrections apply infrequently to historical nodes, leaving the front end to register new data against a drifting prior. This creates a temporal inconsistency between the local and global estimates. Our work addresses this by moving GNSS quality control and frame conversion forward, producing low-latency residuals for immediate front-end updates. This mechanism transforms absolute anchoring from a delayed, historical fix into continuous guidance for ongoing estimation and mapping.

Moreover, unlike backend-centric correction schemes that mainly refine historical poses in the optimizer [8], [13], H-GLIVO closes the correction loop to the active front end. Specifically, the optimized trajectory is used to reconstruct a temporary voxel map and perform registration-refined state correction with atomic map replacement, so that subsequent scan-to-map registration is carried out against a corrected local prior rather than a stale drifting map.

III. METHODOLOGY

A. System Overview

The system is architected into three concurrently running components: the front-end LIVO, an online trajectory alignment module, and a back-end factor-graph smoother. As shown

in Fig. 1, the front end fuses IMU, LiDAR, and visual data to provide high-rate state estimates. The alignment module continuously computes the rigid-body transform between the local odometry frame and the global GNSS ENU frame. This enables two critical data streams: (1) the aligned GNSS measurements are fed back to the front-end IESKF for direct, low-latency correction; and (2) the same GNSS measurements, together with IMU pre-integration and local odometry, are supplied to the back end for high-accuracy smoothing. The back-end optimization then triggers a correction that updates the front-end state and its voxel map, thereby ensuring global consistency.

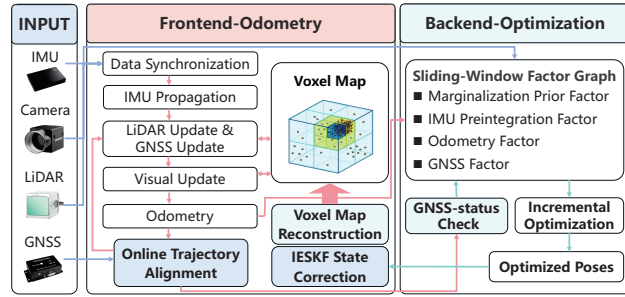


Fig. 1. Overall architecture of the proposed H-GLIVO framework.

B. Front-End LIVO with Direct GNSS Correction

The front end is built upon the robust and efficient FAST-LIVO2 [2] framework, which employs an IESKF to tightly fuse measurements from IMU, LiDAR, and a camera. We extend this framework by directly incorporating GNSS measurements into the filter's update stage.

1) *IESKF Framework*: The system state $\mathbf{x}$ is defined on the composite manifold $\mathcal{M} = SO(3) \times \mathbb{R}^{16}$ with a total dimension of 19:

$$\mathbf{x} = [\mathbf{R} \quad \mathbf{p} \quad \mathbf{v} \quad \mathbf{b}_a \quad \mathbf{b}_g \quad \mathbf{g} \quad t_e^{-1}]^\top \quad (1)$$

where $\mathbf{R} \in SO(3)$ is the orientation, $\mathbf{p} \in \mathbb{R}^3$ is the position, $\mathbf{v}$ is the velocity, and $\mathbf{g}$ is the gravity vector; all are expressed in the world frame $\{W\}$. The terms $\mathbf{b}_a$ and $\mathbf{b}_g$ denote the accelerometer and gyroscope biases, and t_e^{-1} is the estimate of the inverse camera exposure time. These are modeled as random-walk processes.

The filter operates in a standard predict-update cycle. In the prediction step, high-rate IMU measurements are integrated to propagate the nominal state, while the covariance is updated using linearized error-state dynamics. This provides a high-frequency motion prior and is used to deskew LiDAR points. In the update step, LiDAR and camera observations are jointly used to iteratively correct the predicted state. For LiDAR, the deskewed point cloud is matched to a voxelized local map and point-to-plane geometric residuals are formed. For vision, a direct method is adopted: for map points observed in 3D, photometric residuals are constructed by minimizing patch-wise intensity errors between the current and reference image

frames. These residuals constrain the system pose; see FAST-LIVO2 for details. Our main contribution here is to integrate an additional measurement source—absolute GNSS positions.

2) *RTK Measurement Model in IESKF*: The novelty of our front-end IESKF update is that it fuses, within a unified framework, high-rate LiDAR geometric constraints together with low-rate yet globally accurate GNSS position constraints. Once the online trajectory alignment (Sec. III-E) is available, a continuous GNSS position stream transformed into the odometry frame $\{O\}$ is obtained. For a given timestamp k , we have a GNSS position $\mathbf{z}_{\text{gnss},k} \in \mathbb{R}^3$ and its transformed covariance $\mathbf{P}_{\text{gnss},k}$. This measurement directly observes the position component of the state, and the measurement model is:

$$\mathbf{z}_{\text{gnss}} = \mathbf{h}_{\text{gnss}}(\mathbf{x}) + \mathbf{n}_{\text{gnss}}, \quad \mathbf{n}_{\text{gnss}} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{\text{gnss}}) \quad (2)$$

The GNSS residual $\mathbf{r}_{\text{gnss},k}$ is defined as the difference between the measured position $\mathbf{z}_{\text{gnss},k}$ and the predicted state position $\hat{\mathbf{p}}_k$. Its corresponding Jacobian matrix $\mathbf{H}_{\text{gnss}}$ is the partial derivative of the measurement function $\mathbf{h}(\mathbf{x})$ with respect to the error state $\delta\mathbf{x}$:

$$\mathbf{r}_{\text{gnss},k} = \mathbf{z}_{\text{gnss},k} - \hat{\mathbf{p}}_k, \quad \mathbf{H}_{\text{gnss}} = \frac{\partial \mathbf{r}_{\text{gnss},k}}{\partial \delta\mathbf{x}} \quad (3)$$

Before fusion, we apply Mahalanobis-distance gating for robustness against GNSS outliers. The squared Mahalanobis distance is computed as:

$$d_M^2 = \mathbf{r}_{\text{gnss},k}^\top (\mathbf{H}_{\text{gnss}} \mathbf{P}_{k|k-1} \mathbf{H}_{\text{gnss}}^\top + \mathbf{P}_{\text{gnss},k})^{-1} \mathbf{r}_{\text{gnss},k} \quad (4)$$

where $\mathbf{P}_{k|k-1}$ is the predicted state covariance. A measurement is accepted only if $d_M^2 < \chi_{3,1-\alpha}^2$ (the chi-squared threshold), which effectively rejects measurements inconsistent with the high-rate motion prior and protects the front-end IESKF from transient shocks.

During the iterative IESKF update, all LiDAR point-to-plane residuals $\{r_{L,i}\}$ and the accepted GNSS residual $\mathbf{r}_{\text{gnss},k}$ are stacked to form the total residual vector $\mathbf{r}_{\text{total}}$, and their Jacobians are stacked to form $\mathbf{H}_{\text{total}}$:

$$\mathbf{r}_{\text{total}} = \begin{bmatrix} r_{L,1} \\ \vdots \\ r_{L,m} \\ \mathbf{r}_{\text{gnss},k} \end{bmatrix}, \quad \mathbf{H}_{\text{total}} = \begin{bmatrix} \mathbf{H}_{L,1} \\ \vdots \\ \mathbf{H}_{L,m} \\ \mathbf{H}_{\text{gnss}} \end{bmatrix} \quad (5)$$

Since LiDAR and GNSS measurement errors are uncorrelated, the total measurement covariance $\mathbf{R}_{\text{total}}$ is block-diagonal, explicitly encoding our confidence in each measurement:

$$\mathbf{R}_{\text{total}} = \text{diag}(\sigma_{L,1}^2, \dots, \sigma_{L,m}^2, \mathbf{P}_{\text{gnss},k}) \quad (6)$$

Finally, we use the composite measurement model $(\mathbf{r}_{\text{total}}, \mathbf{H}_{\text{total}}, \mathbf{R}_{\text{total}})$ and follow the efficient formulations from FAST-LIO to compute the Kalman gain and state increment, thereby correcting the predicted state.

C. Back-End Optimization for Multi-Sensor Fusion

To obtain a globally consistent and smooth trajectory estimate, we build a back-end optimization framework based on an incremental fixed-lag smoother implemented with GT-SAM [6]. The smoother jointly optimizes the states of a set of keyframes within a sliding time window by solving a large-scale nonlinear least-squares problem.

1) *Factor-Graph Formulation and MAP Estimation:* To achieve a globally consistent estimate of the system state, we formulate multi-sensor fusion as a Maximum a Posteriori (MAP) estimation problem and model it with a factor graph. As illustrated in Fig. 2, our factor-graph model consists of variable nodes representing keyframe states and factors representing different measurement constraints. A factor graph is a bipartite graph composed of variable nodes (unknown states to be optimized) and factor nodes (sensor measurement constraints). In our system, the state at each keyframe k , denoted $\mathbf{x}_k$, consists of the pose $\mathbf{T}_k \in SE(3)$ in the world frame $\{W\}$, the velocity $\mathbf{v}_k \in \mathbb{R}^3$, and the IMU bias $\mathbf{b}_k \in \mathbb{R}^6$. Let $\mathcal{X} = \{\mathbf{x}_k\}_{k \in \mathcal{W}}$ be the collection of all state variables within the current sliding window $\mathcal{W}$.

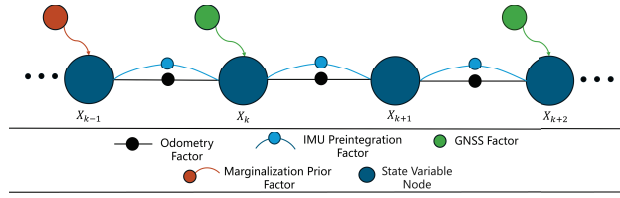


Fig. 2. Factor-graph structure within a fixed-lag window: system states (poses, velocities, IMU biases) connected by IMU, odometry, and GNSS factors plus a marginalized prior.

Our goal is to find the configuration $\mathcal{X}^*$ that maximizes the posterior $p(\mathcal{X} | Z)$ given the measurement set Z and the prior $p(\mathcal{X})$:

$$\mathcal{X}^* = \arg \max_{\mathcal{X}} p(\mathcal{X} | Z) \propto \arg \max_{\mathcal{X}} p(Z | \mathcal{X}) p(\mathcal{X}) \quad (7)$$

Assuming zero-mean Gaussian measurement noises, maximizing the posterior is equivalent to minimizing the sum of negative log-likelihoods over all factors. The MAP problem is thus converted to a nonlinear least-squares objective:

$$\mathcal{X}^* = \arg \min_{\mathcal{X}} \left\{ \|\mathbf{r}_{\text{prior}}(\mathcal{X})\|_{\Sigma_{\text{prior}}}^2 + \sum_{k \in \mathcal{W}} \left(\|\mathbf{r}_{\text{imu}}(\mathbf{x}_{k-1}, \mathbf{x}_k)\|_{\Sigma_{\text{imu}}}^2 + \|\mathbf{r}_{\text{odom}}(\mathbf{T}_{k-1}, \mathbf{T}_k)\|_{\Sigma_{\text{odom}}}^2 + \mathbb{I}(k \in \mathcal{K}_{\text{gnss}}) \|\mathbf{r}_{\text{gnss}}(\mathbf{T}_k)\|_{\Sigma_{\text{gnss}}}^2 \right) \right\} \quad (8)$$

The prior term $\|\mathbf{r}_{\text{prior}}(\mathcal{X})\|_{\Sigma_{\text{prior}}}^2$ is crucial. In our fixed-lag framework, it is not static: to keep a fixed window size for real-time performance, when the oldest variable leaves the window, its information cannot be discarded. We therefore marginalize it, integrating all connected factors into a single marginalized prior that is attached to the remaining “boundary” variables. This passes condensed historical information to the current optimization. Hence, the prior in the objective mainly consists

of this dynamically generated marginalized prior together with an initial prior used at system start-up to anchor the trajectory. The remaining terms are measurement factors: $\mathbf{r}_{\text{imu}}$, $\mathbf{r}_{\text{odom}}$, and $\mathbf{r}_{\text{gnss}}$ denote IMU preintegration, odometry, and GNSS residuals, respectively. The index set $\mathcal{K}_{\text{gnss}} \subseteq \mathcal{W}$ collects keyframes with valid GNSS, and the indicator $\mathbb{I}(\cdot)$ ensures GNSS factors are added only where measurements exist.

2) *IMU Preintegration Factor:* To avoid re-integrating IMU measurements at every iteration, we adopt manifold IMU preintegration. Between consecutive keyframes $k-1$ and k , IMU angular velocity and acceleration are integrated to obtain the relative rotation $\Delta \mathbf{R}_{k,k-1}$, velocity change $\Delta \mathbf{v}_{k,k-1}$, and position change $\Delta \mathbf{p}_{k,k-1}$. These preintegrated quantities are computed in the IMU frame of keyframe $k-1$, independent of the gravity direction and initial pose, and depend only on IMU biases. The residual is:

$$\mathbf{r}_{\text{imu}}(\mathbf{x}_{k-1}, \mathbf{x}_k) = \begin{bmatrix} \log \left(\left(\Delta \hat{\mathbf{R}}_{k,k-1}(\hat{\mathbf{b}}_{k-1}) \right)^T \mathbf{R}_{k-1}^T \mathbf{R}_k \right)^\vee \\ \mathbf{R}_{k-1}^T (\mathbf{v}_k - \mathbf{v}_{k-1} - \mathbf{g} \Delta t) - \Delta \hat{\mathbf{v}}_{k,k-1}(\hat{\mathbf{b}}_{k-1}) \\ \mathbf{R}_{k-1}^T (\mathbf{p}_k - \mathbf{p}_{k-1} - \mathbf{v}_{k-1} \Delta t - \frac{1}{2} \mathbf{g} \Delta t^2) - \Delta \hat{\mathbf{p}}_{k,k-1}(\hat{\mathbf{b}}_{k-1}) \end{bmatrix} \quad (9)$$

3) *Odometry Factor:* The relative pose $\Delta \hat{\mathbf{T}}_{k,k-1}$ from the front-end LIVO is used as an odometry factor constraining $\mathbf{T}_{k-1}$ and $\mathbf{T}_k$. Its residual is defined as:

$$\mathbf{r}_{\text{odom}}(\mathbf{T}_{k-1}, \mathbf{T}_k) = \log \left((\mathbf{T}_{k-1}^{-1} \mathbf{T}_k) \Delta \hat{\mathbf{T}}_{k,k-1}^{-1} \right)^\vee \in \mathfrak{se}(3) \quad (10)$$

To enhance robustness against occasional failures or large drifts in the front-end odometry, a robust loss $\rho(\cdot)$ (e.g., Cauchy or Huber) is applied so that the weight of this factor is downweighted automatically when the residual norm is large.

4) *GNSS Factor and Robust Screening:* In the back end, the aligned GNSS measurements are instantiated as unary GNSS factors to provide long-term global position constraints. The residual connects the translation of $\mathbf{T}_k$ to the GNSS position $\mathbf{z}_{\text{gnss},k}$:

$$\mathbf{r}_{\text{gnss}}(\mathbf{T}_k) = \text{translation}(\mathbf{T}_k) - \mathbf{z}_{\text{gnss},k} \quad (11)$$

GNSS quality is highly sensitive to harsh environments (urban canyons, tunnels, dense foliage), producing non-Gaussian noise and outliers. A robust loss alone is insufficient; strict screening must precede factor insertion. We therefore employ a multi-stage pipeline—pre-checks on time sync, minimum travel distance since the last valid GNSS, and reported covariance thresholds—followed by Mahalanobis-distance chi-squared gating. The squared Mahalanobis distance of the GNSS residual against its innovation covariance is:

$$d_M^2 = \mathbf{r}_{\text{gnss},k}^\top \left(\mathbf{H}_{\text{gnss}} \mathbf{P}_{k|k}^{\text{marg}} \mathbf{H}_{\text{gnss}}^\top + \mathbf{P}_{\text{gnss},k} \right)^{-1} \mathbf{r}_{\text{gnss},k} \quad (12)$$

where $\mathbf{P}_{k|k}^{\text{marg}}$ is the marginalized posterior covariance queried from the smoother (more accurate and better conditioned than the predictive covariance used in the front-end IESKF). The value d_M^2 approximately follows χ_3^2 ; a measurement is accepted as an inlier only if d_M^2 falls below a preset threshold (e.g., 11.345 for a 99% confidence interval), after which the

GNSS factor is added to the graph. This effectively rejects GNSS outliers (e.g., due to multipath) and is key to a robust back-end optimization.

D. Back-End-Guided Front-End Correction

A core innovation of our system is a back-end-guided correction mechanism that feeds the back-end global optimization results to the front end, proactively eliminating accumulated drift and ensuring long-term consistency. The procedure consists of two key stages: back-end-guided map reconstruction, and registration-refined state correction with an atomic update.

1) *Back-End-Guided Map Reconstruction*: This procedure is executed asynchronously by a dedicated background thread to avoid blocking the real-time front end. When the back-end smoother (e.g., after receiving GNSS corrections) computes an updated segment of trajectory, the thread is triggered. It first retrieves the optimized poses within a time window (e.g., the most recent N keyframes), $\{\mathbf{T}_{W,I_i}^*\}$, together with their raw historical point clouds $\{\mathcal{P}_{L,i}\}$. By reprojecting historical point clouds using these globally consistent optimized poses, a new, geometrically more accurate temporary voxel map $\mathcal{M}_{\text{temp}}$ is reconstructed:

$$\mathcal{M}_{\text{temp}} = \text{BuildVoxelMap}\left(\bigcup_{i=k-N}^k \{\mathbf{T}_{W,I_i}^* (\mathbf{R}_{IL} \mathbf{p}_j + \mathbf{t}_{IL}) \mid \mathbf{p}_j \in \mathcal{P}_{L,i}\}\right) \quad (13)$$

This reconstruction effectively removes geometric distortions in the voxel map caused by front-end drift, yielding a small-area yet high-precision map.

2) *IESKF State Correction and Atomic Map Exchange*: Simply applying a correction to the current state is insufficient, because it cannot fully resolve the mismatch between the current frame and the new map. We therefore adopt a registration-refinement strategy to precisely correct the current state. We compute a coarse correction transform $\Delta\mathbf{T}_{\text{corr}}(\Delta\mathbf{R}_{\text{corr}}, \Delta\mathbf{p}_{\text{corr}})$ representing the difference between the original front-end pose $\mathbf{T}_{W,I_k}$ and the back-end optimized pose $\mathbf{T}_{W,I_k}^*$ at a recent keyframe k :

$$\Delta\mathbf{T}_{\text{corr}} = \mathbf{T}_{W,I_k}^* (\mathbf{T}_{W,I_k})^{-1} \quad (14)$$

We then apply this transform to the latest front-end pose and velocity estimates, $\hat{\mathbf{T}}_{\text{curr}}$ and $\hat{\mathbf{v}}_{\text{curr}}$, obtaining the initial guess for entering the new map:

$$\begin{aligned} \hat{\mathbf{T}}_{\text{curr,guess}} &= \Delta\mathbf{T}_{\text{corr}} \hat{\mathbf{T}}_{\text{curr}}, \\ \hat{\mathbf{v}}_{\text{curr,guess}} &= \Delta\mathbf{R}_{\text{corr}} \hat{\mathbf{v}}_{\text{curr}} \end{aligned} \quad (15)$$

The critical step is to register the current live LiDAR scan $\mathcal{P}_{\text{curr}}$ to the newly reconstructed, high-accuracy temporary map $\mathcal{M}_{\text{temp}}$. Starting from $\hat{\mathbf{T}}_{\text{curr,guess}}$, the scan-to-map registration refines the pose to a final estimate $\mathbf{T}_{\text{curr}}^*$ precisely aligned with the new map. For thread safety and seamless switching, we perform an atomic update: the front end is briefly locked, the active map pointer is redirected to $\mathcal{M}_{\text{temp}}$, and the front-end state is reset to the refined pose $\mathbf{T}_{\text{curr}}^*$.

This reconstruct-register-replace correction pipeline actively pulls the front end back to a globally consistent reference frame, fundamentally removing accumulated errors and ensuring integrity and accuracy during long-term operation.

E. Online Trajectory Alignment

To fuse the global position information from GNSS, we designed an online trajectory alignment module to estimate and compensate for the unknown transformation between the local odometry frame $\{O\}$ and the global GNSS local ENU frame $\{E\}$. This module first establishes a fixed ENU reference frame using the *GeographicLib* library, anchored at the first received GNSS point. Subsequently, by analyzing the displacement vectors of the two trajectories (odometry and GNSS) during the initial motion phase of the system, it analytically computes the heading deviation (rotation) and origin offset (translation) between them in a single step. Once this transformation is determined, all subsequent GNSS measurements and their associated uncertainties are transformed into the odometry frame in real-time. This allows them to be seamlessly integrated as direct observations into both the front-end IESKF and the back-end factor graph optimization, ensuring consistency between global and local information.

IV. EXPERIMENTS

To comprehensively evaluate the performance of our system, we conduct a series of quantitative and qualitative experiments. We first describe the experimental setup, and then validate the superiority of our approach and the effectiveness of its key modules through quantitative comparisons against strong baselines and ablation studies. Finally, we demonstrate qualitative results and practical value by deploying our system in large-scale real-world scenarios.

A. Experimental Setup

Our experiments are carried out on a self-built portable mobile mapping platform (see Fig. 3), which integrates a Livox Avia LiDAR, a Hikvision industrial camera, and an external high-precision RTK-GNSS system. All data are processed in real time on an Intel i9-13900H host. We evaluate on three public datasets—Railroad-dataset [18], M2DGR [19], and M3DGR [20]—as well as a private dataset covering approximately 1.4 km of a complex campus environment. These datasets span diverse and challenging scenarios from long-range railways and tunnels to dynamic scenes and weak textures.

For performance comparison, we benchmark our full method (Ours) against several advanced baselines: FAST-LIO2 [1], LIO-SAM [8], and FAST-LIVO2 [2]. For fairness, FAST-LIO2 and LIO-SAM are both augmented with a unified, simple loosely coupled GNSS module, denoted as FAST-LIO2-GPS and LIO-SAM-GPS. We also design two ablated variants to validate the internal components of our method: Ours w/o Map Correction (disabling the back-end to front-end correction feedback) and Ours w/o Outlier Rejection (removing the robust screening mechanism for GNSS measurements). The primary quantitative metric is the root mean square error (RMSE) of the Absolute Trajectory Error (ATE).

B. Quantitative Evaluation and Ablation Analysis

All ATE RMSE results on public datasets are summarized in Table I.

TABLE I
ABSOLUTE TRANSLATIONAL ERROR (ATE) RMSE [M] ACROSS PUBLIC DATASETS

Dataset	Sequence	Dist. (m)	FAST-LIO2-GPS	LIO-SAM-GPS	FAST-LIVO2	Ours w/o Map Corr.	Ours w/o Outlier Rej.	Ours (Full)
Railroad-dataset	no_tunnel	5448.6	4.037	×	130.055	<u>2.565</u>	8.565	0.513
	2021-11-11-13-49-48	3091.9	10.667	×	179.430	<u>3.232</u>	6.156	0.578
	two_tunnels	3494.0	15.881	×	124.902	<u>2.745</u>	8.492	2.106
M2DGR	door_01	142.8	0.484	0.685	<u>0.425</u>	0.768	2.335	0.371
	gate_03	248.4	1.049	0.815	<u>0.309</u>	1.203	1.478	0.289
	walk_01	263.2	1.887	0.777	<u>0.511</u>	0.846	1.565	0.478
M3DGR	Outdoor01	345.8	<u>0.573</u>	×	0.235	0.923	0.965	0.817
	Outdoor04	680.4	×	1.573	0.875	1.428	1.563	1.373
	Dynamic04	332.7	2.002	×	1.073	1.592	1.356	<u>1.175</u>

Dist. (m) denotes sequence length. **Bold** and underline mark the first and second best results per row, respectively. × indicates failure or unavailable result.

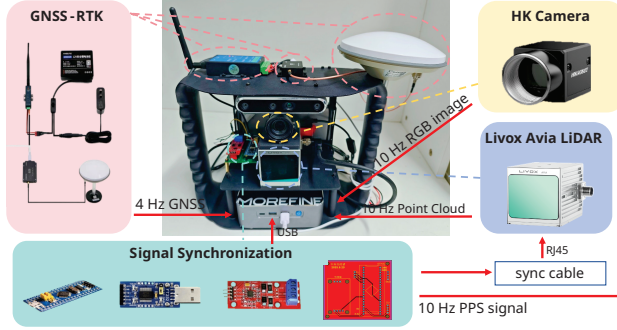


Fig. 3. Our portable mapping platform integrating a Livox Avia LiDAR, a Hikvision industrial camera, and a high-precision GNSS-RTK unit, with real-time processing on an Intel i9-13900H host.

1) *Performance under large-scale and GNSS-challenging scenarios:* On the most challenging Railroad-dataset, the advantages of our method are fully demonstrated. This dataset simulates long-range inspection tasks across mixed environments (open areas and tunnels), posing a stringent test for the long-term consistency and global localization capability of SLAM systems.

On the 5.4 km *no_tunnel* sequence, our full method (Ours) achieves an ATE RMSE of only 0.513 m, representing a significant improvement over the simple loosely coupled FAST-LIO2-GPS (4.037 m) and the severely drifting FAST-LIVO2 (130.055 m) due to the lack of global constraints. This result strongly indicates that, for large-scale applications, our back-end fusion strategy with GNSS is an effective route to high-accuracy global localization.

Ablation experiments further reveal the key mechanisms behind this superior performance. First, *Ours w/o Map Correction* (ATE 2.565 m) already performs markedly better than pure front-end methods, suggesting that the back-end factor graph alone can recover the globally correct *shape* of the trajectory. However, its notable gap from our full method (0.513 m) demonstrates that the active correction feedback from the back end to the front end is decisive for fundamentally eliminating front-end accumulation error and unifying the states of the front and back ends. Second, the failure of *Ours w/o Outlier Rejection* (ATE 8.565 m) highlights the absolute necessity of

robust screening. As shown in Fig. 4, without effectively rejecting anomalous GNSS measurements, erroneous global information catastrophically corrupts the factor graph and leads to divergence. This confirms that our Mahalanobis-distance-based gating strategy is a prerequisite for successful GNSS fusion.

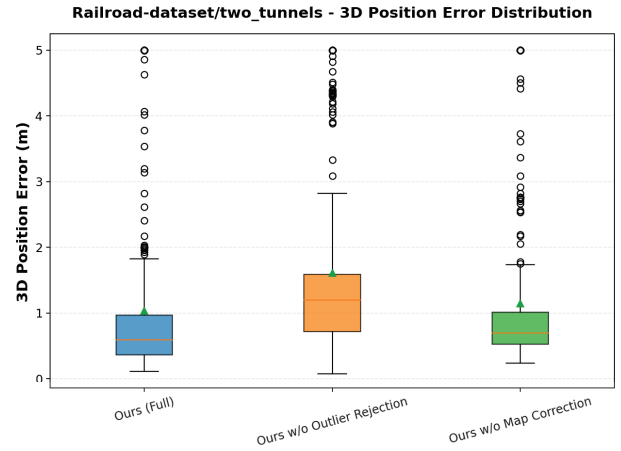


Fig. 4. Ablation of GNSS fusion: distribution of 3D position error. The full system exhibits the tightest error distribution with the lowest median and mean, while removing outlier rejection leads to a wider spread with numerous outliers, and removing map correction results in a slightly degraded but still bounded distribution.

In the *two_tunnels* sequence, GNSS signals are completely lost for extended periods. Under such extreme conditions, LIO-SAM rapidly diverges after losing GNSS constraints, whereas our method not only maintains a stable trajectory estimate but also achieves a final ATE of 2.106 m. This is because our system degrades effectively to a high-performance LiDAR-inertial odometry during GNSS outages and, upon signal recovery, uses robust screening and active correction feedback to pull the drifted tunnel trajectory back to the globally correct position, thereby ensuring long-term stability after GNSS returns.

2) *Competitiveness on general SLAM datasets:* To verify the generality of our framework, we further evaluate it on two widely used SLAM benchmarks, M2DGR and M3DGR. On M2DGR, where GNSS signals are relatively weak, our method

attains state-of-the-art accuracy (e.g., on the *gate_03* and *walk_01* sequences) by jointly smoothing IMU and odometry factors and applying global corrections using sparse yet valid GNSS measurements.

On M2DGR, our approach exhibits strong competitiveness; for instance, it achieves the best accuracy on both *gate_03* and *walk_01*. This highlights the advantage of our back-end framework under weak GNSS constraints. Rather than simply relying on GNSS, it continuously performs joint smoothing of IMU pre-integration factors and front-end odometry factors to construct a high-precision, internally consistent relative trajectory. When a sparse but reliable GNSS measurement passes our robust screening, it can effectively provide global correction to this high-quality relative trajectory. This demonstrates the adaptability of our multi-sensor fusion strategy, which leverages all available information sources to deliver high-accuracy localization, outperforming standalone LIO or LIVO systems.

Notably, on certain short M3DGR sequences (e.g., *Outdoor01*), our error is slightly higher than that of the pure front end FAST-LIVO2. We attribute this mainly to the fact that drift is not yet significant over short trajectories, and that the GNSS in this dataset lacks informative covariance, making it challenging for our back end to filter low-quality measurements and introducing mild perturbations to an otherwise locally precise trajectory. This observation also underscores the importance of accurate measurement uncertainty for achieving optimal sensor fusion.

C. Qualitative Evaluation

To intuitively demonstrate the performance of our system, we visualize the optimized trajectories and the reconstructed point-cloud maps.

As shown in Fig. 5, on the most challenging *Railroad/two_tunnels* sequence, our method (red trajectory) closely matches the ground truth (black dashed) along all axes. Even after prolonged GNSS outages caused by long tunnels, the trajectory rapidly reconverges to the ground truth once the signal resumes, whereas other methods exhibit significant divergence or drift. To further verify global consistency, we project the point-cloud map of this sequence onto satellite imagery, as shown in Fig. 6. The resulting map aligns precisely with the railway alignment in the satellite image, demonstrating excellent georeferencing accuracy.

Finally, we deploy our system on a private campus dataset spanning 1.4 km to assess its end performance in the real world. As shown in Fig. 7(a), the left shows a front-end-only method and the right shows our full method; our approach produces a globally consistent, distortion-free colored point cloud. Thanks to back-end global optimization and correction feedback to the front end, building contours are sharp, road edges are crisp, and there is no ghosting even in loop-closure areas. In contrast, the front-end-only method suffers from significant cumulative error, producing visible ghosting and geometric distortion in loop regions. Moreover, Fig. 7(b) overlays the reconstructed map and trajectories onto satellite

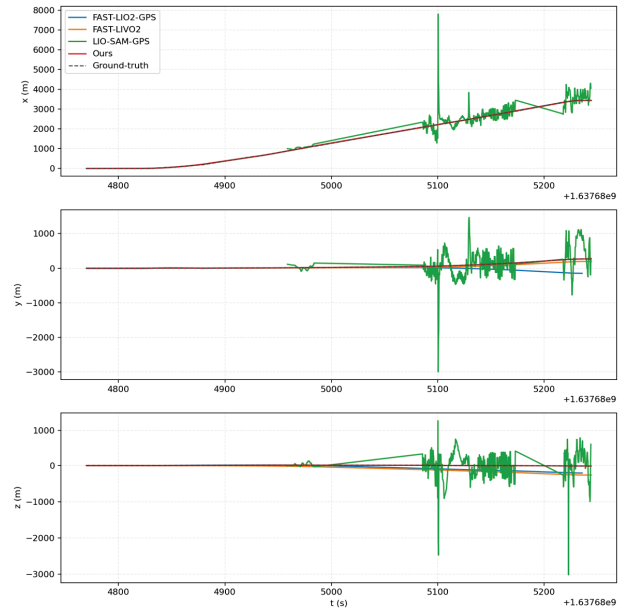


Fig. 5. Railroad/two_tunnels XYZ trajectory comparison. The proposed method (red) closely matches the ground truth (black dashed) along all axes and rapidly reconverges after GNSS recovery, while competing methods drift or diverge.

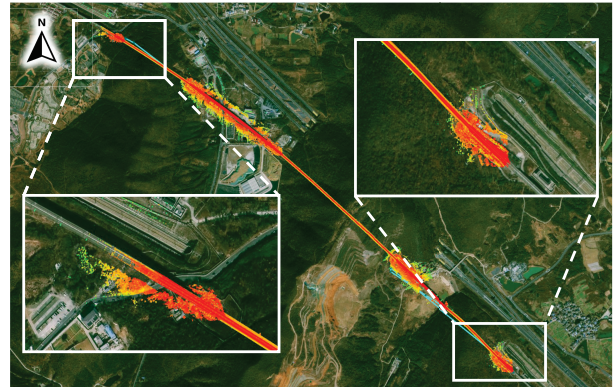


Fig. 6. Railroad tunnel scene: reconstructed point-cloud map projected onto satellite imagery. The map aligns precisely with the railway alignment, demonstrating high georeferencing accuracy.

imagery. The blue trace illustrates the drifting path estimated by the standalone front-end odometry, while the red trace denotes the globally corrected trajectory derived from the back-end optimization. The precise alignment of the red trajectory with the satellite base map further confirms the high-accuracy georegistration of our results.

V. CONCLUSION

This paper presented a robust and high-accuracy H-GLIVO SLAM framework. The core innovation lies in enabling bidirectional information flow between a real-time front-end

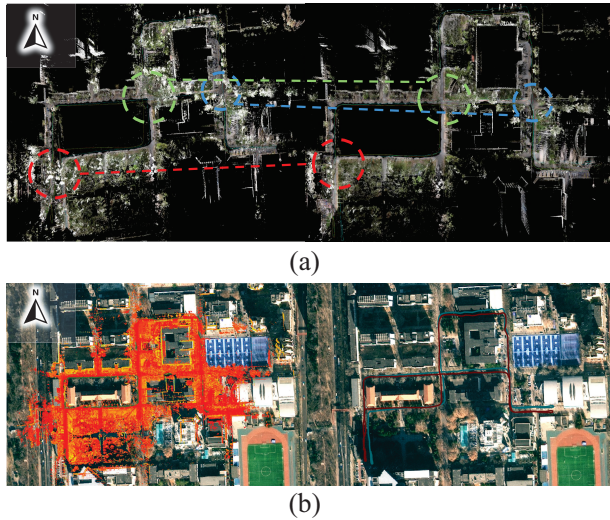


Fig. 7. Private-campus dataset: (a) Globally consistent, ghosting-free colored point-cloud map with highlighted loop/repetitive regions; (b) Satellite overlay of map and trajectory verifying accurate georegistration.

odometry and a back-end global smoother. Through a back-end-guided correction mechanism, the framework leverages the globally optimized trajectory to asynchronously reconstruct the front-end map and rectify the front-end state. This active feedback fundamentally eliminates accumulated drift in the front end and ensures long-term global consistency. Extensive experiments on multiple public and private large-scale datasets validate the effectiveness of the proposed method. Quantitative results show substantial accuracy gains over baseline systems in long-range and GNSS-challenging scenarios. Ablation studies further confirm the necessity of our robust GNSS screening strategy and the back-end-to-front-end correction feedback for achieving superior performance.

REFERENCES

- [1] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-liv2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [2] C. Zheng, W. Xu, Z. Zou, T. Hua, C. Yuan, D. He, B. Zhou, Z. Liu, J. Lin, F. Zhu, Y. Ren, R. Wang, F. Meng, and F. Zhang, "Fast-liv2: Fast, direct lidar-inertial-visual odometry," *IEEE Transactions on Robotics*, vol. 41, pp. 326–346, 2025.
- [3] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5692–5698.
- [4] J. Lin and F. Zhang, "R3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 672–10 678.
- [5] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3281–3288.
- [6] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," *Georgia Institute of Technology*, 2012.
- [7] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.
- [8] 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.
- [9] X. Li, H. Yu, X. Wang, S. Li, Y. Zhou, and H. Chang, "Fgo-gil: Factor graph optimization-based gnss rtk/ins/lidar tightly coupled integration for precise and continuous navigation," *IEEE Sensors Journal*, vol. 23, no. 13, pp. 14 534–14 548, 2023.
- [10] T. Li, L. Pei, Y. Xiang, X. Zuo, W. Yu, and T.-K. Truong, "P3-lins: Tightly coupled ppp-gnss/ins/lidar navigation system with effective initialization," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.
- [11] X. Liu, W. Wen, and L.-T. Hsu, "Glio: Tightly-coupled gnss/lidar/imu integration for continuous and drift-free state estimation of intelligent vehicles in urban areas," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1412–1422, 2024.
- [12] X. Li, H. Wang, S. Li, S. Feng, X. Wang, and J. Liao, "Gil: A tightly coupled gnss ppp/ins/lidar method for precise vehicle navigation," *Satellite navigation*, vol. 2, no. 1, p. 26, 2021.
- [13] W. Wu, X. Zhong, D. Wu, B. Chen, X. Zhong, and Q. Liu, "Lio-fusion: Reinforced lidar inertial odometry by effective fusion with gnss/relocalization and wheel odometry," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1571–1578, 2023.
- [14] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.
- [15] C. Yuan, J. Lin, Z. Zou, X. Hong, and F. Zhang, "Std: Stable triangle descriptor for 3d place recognition," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1897–1903.
- [16] C. Yuan, J. Lin, Z. Liu, H. Wei, X. Hong, and F. Zhang, "Btc: A binary and triangle combined descriptor for 3-d place recognition," *IEEE Transactions on Robotics*, vol. 40, pp. 1580–1599, 2024.
- [17] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1879–1884.
- [18] Y. Wang, Y. Lou, W. Song, Z. Tu, Y. Wang, and S. Zhang, "Simultaneous localization of rail vehicles and mapping of surroundings with lidar-inertial-gnss integration," *IEEE Sensors Journal*, vol. 22, no. 14, pp. 14 501–14 512, 2022.
- [19] 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*, vol. 7, no. 2, pp. 2266–2273, 2022.
- [20] D. Zhang, J. Zhang, Y. Sun, T. Li, H. Yin, H. Xie, and J. Yin, "Towards robust sensor-fusion ground slam: A comprehensive benchmark and a resilient framework," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 8894–8901.