Supplementary Material for "Reliability-Aware LiDAR–RGB-D–IMU Fusion for Dense Local Mapping and Odometry in Autonomous Ground Robots"

Workagegn Tatek Asfu[a], Shoukun Wang[a,*], Zhenhai Long[a]

[a]*School of Automation, Beijing Institute of Technology, Beijing, China*

Abstract

This supplementary material supports the article "Reliability-Aware LiDAR–RGB-D–IMU Fusion for Dense Local Mapping and Odometry in Autonomous Ground Robots." It provides implementation details, pseudocode, parameter settings, dataset and evaluation notes, runtime information, and additional qualitative figures for the reliability-aware LiDAR–RGB-D–IMU fusion system. The document follows the terminology of the main article: bidirectional diffusion synchronization (BDS), Prediction Uncertainty Tensor (PUT), diffusion-augmented multi-modal calibration (DAMC), SE(3)-consistent feature fusion, diffusion-informed Kalman filtering (DIKF), and predictive occupancy field (POF) dense mapping. The added supplementary figures show the PUT construction stage and an extended LiDAR odometry trajectory visualization.

Keywords: Supplementary material, dense local mapping, LiDAR–RGB-D–IMU fusion, reliability-aware sensor fusion, uncertainty-aware odometry, predictive occupancy field

S1. Scope of the Supplementary Material

This document is intended to make the implementation and evaluation protocol easier to reproduce without changing the claims of the main article. It provides:

*Corresponding author.

Email addresses: workagegntatek@gmail.com (Workagegn Tatek Asfu), bitwsk@bit.edu.cn (Shoukun Wang), longzhh@bit.edu.cn (Zhenhai Long)

Table S1: Reported Sensor Configuration for the Campus Platform

Sensor	**Reported configuration**
LiDAR	Two RoboSense Helios 16-channel scanners; each operated at 10 Hz; approximately 10^5 points per frame; maximum range about 100 m; dual layout for wide horizontal coverage.
RGB-D camera	Intel RealSense D435i; RGB-D stream at 640×480 resolution and 30 Hz.
IMU	LPMS-IG1 9-DOF IMU; sampling rate 400 Hz.
Synchronization	LiDAR timestamps define the main evaluation clock; RGB-D frames are associated with the nearest LiDAR timestamp within a fixed tolerance; IMU samples are integrated between consecutive LiDAR frames.

- the reported hardware and software configuration used for the campus experiments;
- pseudocode for the main system components;
- parameter settings for synchronization, uncertainty prediction, calibration, filtering, and mapping;
- dataset and evaluation notes that distinguish the campus evaluation from auxiliary M2UD uncertainty analysis;
- supplementary qualitative figures for PUT construction and trajectory inspection.

The supplementary trajectory figure included here visualizes one processed replay segment. It should not be interpreted as the complete length of the collected campus dataset. The complete reported campus dataset consists of three sequences totaling 1135 m and 25.4 min, as summarized in Table S7.

S2. Experimental Platform and Software Configuration

S2.1. Sensor Suite

The ground-robot platform used in the reported campus experiments integrates dual LiDAR, RGB-D, and inertial sensing. Hardware triggering provides the main clock reference, while BDS handles residual latency and unequal sampling rates.

Table S2: Software Environment Used for Implementation

Component	**Configuration**
Operating system	Ubuntu 22.04 LTS
Robot middleware	ROS 2 Humble
Python	3.10 series
Deep learning backend	CUDA-enabled PyTorch
Numerical stack	NumPy, SciPy, Eigen
Point-cloud processing	PCL, Open3D
Vision processing	OpenCV
Optimization support	Ceres Solver, g2o-compatible back-end utilities
GPU acceleration	CUDA-enabled inference for RDE, PDP, and feature extraction

S2.2. Computing Platform

All reported campus experiments were processed on an NVIDIA GeForce RTX 4060 GPU and an Intel Core i9 workstation. The DIKF update runs at the 10 Hz LiDAR frame rate. The measured end-to-end latency is 42 ms per frame, which is below the 100 ms LiDAR period. Camera frames are processed in parallel with the LiDAR-rate estimator.

S3. System Pipeline and Algorithmic Details

The proposed system is organized as six processing stages. Stage A constructs the PUT. Stage B performs BDS using the RDE and PDP streams. Stage C refines LiDAR–camera alignment through DAMC. Stage D extracts SE(3)-consistent reliability-weighted features. Stage E performs covariance-adaptive odometry using DIKF. Stage F integrates reliable geometry and appearance into the POF dense local map.

S3.1. Main Reliability-Aware Fusion Pipeline

S3.2. Prediction Uncertainty Tensor Construction

The PUT is a shared voxel representation that stores geometry, RGB-D appearance, IMU motion cues, and uncertainty channels. At each LiDAR evaluation time, modality-specific lifting operators map the measurements into a common voxel coordinate system:

$$\Psi_k = \mathrm{Concat}_C \left(\Phi_{\mathrm{cam}}(I_k, D_k, N_k, S_k), \Phi_{\mathrm{lidar}}(P_k, I_k^L), \Phi_{\mathrm{imu}}(a_k, \omega_k), U_{\mathrm{pred}}(\cdot, t_k + \tau) \right), \tag{S1}$$

Algorithm S1 Reliability-Aware LiDAR–RGB-D–IMU Fusion Pipeline

Input: LiDAR scans P_k, RGB-D frames (I_k, D_k), IMU samples u_i, calibration prior T_{LC}^0
Output: Pose estimate $\hat{x}_k$, covariance P_k, dense map $\mathcal{M}_k$, reliability field U_{pred}

Initialize nominal state $\hat{x}_0$, covariance P_0, local map $\mathcal{M}_0$, and calibration prior T_{LC}^0
for each LiDAR evaluation time t_k **do**
    Associate RGB-D frame to t_k and integrate IMU samples between t_{k-1} and t_k
    // Stage A: PUT construction
    $\Psi_k \leftarrow \text{CreatePUT}(I_k, D_k, P_k, u_{t_{k-1}:t_k})$
    // Stage B: BDS synchronization and reliability prediction
    $(\Psi_k^{\text{sync}}, U_{\text{pred}}^{(k)}, \epsilon_{\theta,k}) \leftarrow \text{BDS}(\Psi_k)$
    // Stage C: reliability-aware LiDAR–camera calibration
    $T_{LC,k} \leftarrow \text{DAMC}(\Psi_k^{\text{sync}}, T_{LC,k-1}, U_{\text{pred}}^{(k)})$
    // Stage D: SE(3)-consistent feature fusion
    $F_k^{\text{multi}} \leftarrow \text{SE3FeatureFusion}(\Psi_k^{\text{sync}}, T_{LC,k}, U_{\text{pred}}^{(k)})$
    // Stage E: diffusion-informed filtering
    $(\hat{x}_k, P_k) \leftarrow \text{DIKF}(\hat{x}_{k-1}, P_{k-1}, u_{t_{k-1}:t_k}, F_k^{\text{multi}}, U_{\text{pred}}^{(k)}, \epsilon_{\theta,k})$
    // Stage F: dense local mapping
    $\mathcal{M}_k \leftarrow \text{POF}(\mathcal{M}_{k-1}, F_k^{\text{multi}}, \hat{x}_k, P_k, U_{\text{pred}}^{(k)})$
end for
return $\{\hat{x}_k, P_k, \mathcal{M}_k, U_{\text{pred}}^{(k)}\}_{k=1}^{N}$

where τ is the prediction horizon. In the submitted configuration, $\tau = 0.1\,\text{s}$, corresponding to one LiDAR period at 10 Hz.

The effective uncertainty used by downstream modules is computed from epistemic, aleatoric, and bounded predictive components:

$$U_{\text{eff}} = \sqrt{U_{\text{epistemic}}^2 + U_{\text{aleatoric}}^2 + \left(\text{clip}(U_{\text{pred}}, 0, U_{\max})\right)^2}. \tag{S2}$$

When U_{eff} increases, the DIKF increases the effective measurement covariance and the POF reduces the influence of the corresponding voxel updates.

S3.3. Bidirectional Diffusion Synchronization

BDS models residual temporal asynchrony as structured noise. The RDE denoises current observations, while the PDP predicts short-horizon reliability.

Algorithm S2 Bidirectional Diffusion Synchronization

Input: PUT Ψ_k, base noise schedule $\{\beta_t\}_{t=1}^{T}$
Output: Synchronized tensor Ψ_k^{sync}, denoising residual $\epsilon_{\theta,k}$, predictive reliability $U_{\text{pred}}^{(k)}$
Extract modality-level latent variables $x_0^{\text{all}} = [x_0^{\text{rgbd}}, x_0^{\text{imu}}, x_0^{\text{lidar}}]^\top$
Estimate compact cross-modal covariance Σ_{cross} from Ψ_k
for $t = 1$ to T **do**
    Predict short-horizon reliability $U_{\text{pred}}(t)$ using PDP
    Set $\beta_t' \leftarrow \text{clip}(\beta_t(1 + U_{\text{pred}}(t)), \beta_{\min}, \beta_{\max})$
    Sample structured noise $\epsilon_t \sim \mathcal{N}(0, \Sigma_{\text{cross}})$
    Apply forward diffusion to modality latent variables
end for
for $t = T$ down to 1 **do**
    Estimate denoising residual $\epsilon_\theta(x_t, t, \Psi_k)$ using RDE
    Recover x_{t-1} using the reverse diffusion mean and covariance
end for
Lift recovered latent variables back to the shared voxel representation
return Ψ_k^{sync}, $\epsilon_{\theta,k}$, $U_{\text{pred}}^{(k)}$

The adaptive diffusion schedule is

$$\beta_t' = \text{clip}\left(\beta_t \left(1 + U_{\text{pred}}(t)\right), \beta_{\min}, \beta_{\max}\right). \tag{S3}$$

S3.4. Diffusion-Augmented Multi-Modal Calibration

DAMC refines the LiDAR–camera transform online. The calibration update is slowed when the predicted uncertainty is high, which reduces the influence of unreliable LiDAR–camera correspondences.

S3.5. SE(3)-Consistent Feature Fusion

After synchronization and calibration, RGB-D and LiDAR features are fused in the common map frame. A variance-based weight balances the RGB-D and LiDAR contributions:

$$\alpha_k = \frac{\sigma_{\text{lidar},k}^2}{\sigma_{\text{rgbd},k}^2 + \sigma_{\text{lidar},k}^2 + \mu}, \tag{S4}$$

where μ is a small numerical constant. A higher LiDAR variance increases the RGB-D contribution, while a higher RGB-D variance increases the LiDAR

Algorithm S3 Reliability-Aware DAMC Update

Input: LiDAR points P^L, RGB-D/depth correspondences P^C, prior transform T^0_{LC}, predicted uncertainty U_{pred}
Output: Updated transform T^*_{LC}
Decompose T^0_{LC} into Cayley rotation parameter $\omega^{(0)}$ and translation $p^{(0)}$
for $j = 1$ to $K_{\max}$ **do**
  Reconstruct $T^{(j)}_{LC}$ from $(\omega^{(j)}, p^{(j)})$
  Project transformed LiDAR points into the RGB-D frame
  Compute correspondence residual $r^{(j)}$ and Jacobian $J^{(j)}$
  Set step size $\eta^{(j)} \leftarrow \eta_0/(1+\sigma^2_{\text{pred}})$
  Solve damped Gauss–Newton step $\delta \leftarrow -\left((J^{(j)})^\top J^{(j)} + \lambda I\right)^{-1} (J^{(j)})^\top r^{(j)}$
  Update $[\omega^{(j+1)}, p^{(j+1)}]^\top \leftarrow [\omega^{(j)}, p^{(j)}]^\top + \eta^{(j)}\delta$
  if $\|\delta\| < \epsilon_{\text{tol}}$ **then**
    break
  end if
end for
return $T^*_{LC} = T^{(j)}_{LC}$

contribution. The resulting multi-modal feature tensor is

$$F_k^{\text{multi}} = \text{Concat}\left(\text{SE3Conv}([F_k^{\text{rgb}}, F_k^{\text{depth}}, F_k^{\text{lidar}}]), D_{\theta,k}^{\text{feature}}, U_{\text{pred}}^{(k)}\right). \tag{S5}$$

S3.6. Diffusion-Informed Kalman Filtering

The DIKF maintains the nominal error-state

$$x_k = (R_k, p_k, v_k, b_{a,k}, b_{g,k}), \tag{S6}$$

where $R_k \in \text{SO}(3)$, $p_k \in \mathbb{R}^3$, $v_k \in \mathbb{R}^3$, and $b_{a,k}$ and $b_{g,k}$ are accelerometer and gyroscope biases. Diffusion-derived covariances are added to the base process and measurement covariances:

$$Q_k = Q_{\text{base}} + \Sigma^Q_\theta(U^{(k)}_{\text{pred}}), \tag{S7}$$

$$R^{\text{eff}}_k = R_{\text{base}} + \Sigma^R_\theta(\epsilon_{\theta,k}) + \Sigma_\theta(k). \tag{S8}$$

The Kalman gain is then computed as

$$K_k = P_{k|k-1}H_k^\top \left(H_k P_{k|k-1} H_k^\top + R_k^{\text{eff}}\right)^{-1}. \tag{S9}$$

Algorithm S4 DIKF Prediction and Update

Input: Prior $\hat{x}_{k-1}$, covariance P_{k-1}, IMU data u, features F_k^{multi}, $U_{\text{pred}}^{(k)}$, denoising residual $\epsilon_{\theta,k}$
Output: Updated state $\hat{x}_k$ and covariance P_k
Propagate IMU state on SE(3) to obtain $\hat{x}_{k|k-1}$
Compute $Q_k \leftarrow Q_{\text{base}} + \Sigma_\theta^Q(U_{\text{pred}}^{(k)})$
Propagate covariance $P_{k|k-1}$
Extract scan/map or feature residual ν_k and measurement Jacobian H_k
Compute $R_k^{\text{eff}} \leftarrow R_{\text{base}} + \Sigma_\theta^R(\epsilon_{\theta,k}) + \Sigma_\theta(k)$
Compute innovation covariance $S_k \leftarrow H_k P_{k|k-1} H_k^\top + R_k^{\text{eff}}$
Compute Kalman gain $K_k \leftarrow P_{k|k-1} H_k^\top S_k^{-1}$
Apply manifold correction to $\hat{x}_{k|k-1}$ using $K_k \nu_k$
Update covariance with Joseph or equivalent numerically stable form
Check normalized innovation squared $\text{NIS}_k = \nu_k^\top S_k^{-1} \nu_k$
if NIS_k is above the selected chi-square gate **then**
    Increase R_k^{eff} or reject the affected residual block
end if
return $\hat{x}_k$, P_k

S3.7. Predictive Occupancy Field Mapping

The POF integrates the state estimate, feature tensor, and PUT uncertainty into a dense local map. The kernel used for occupancy regression is

$$K_{\text{multi}}(x, x') = K_{\text{rgb}}(x, x') K_{\text{depth}}(x, x') K_{\text{lidar}}(x, x') + K_{\text{diff}}(x, x', t) + \delta_{xx'} \sigma_{\text{pred}}^2(x, t). \tag{S10}$$

The diagonal predictive-uncertainty term inflates uncertainty in unreliable regions while preserving positive semi-definiteness.

S4. Parameter Settings

Table S3 summarizes the main synchronization and uncertainty parameters used by the submitted configuration. Table S4 gives the learning settings shared by the BDS, feature, and filtering components.

S5. Dataset, Evaluation Protocol, and Reproducibility Notes

The custom campus dataset is used for trajectory estimation, baseline comparison, ablation, dense mapping, calibration assessment, and runtime

Algorithm S5 POF Dense Local Map Update

Input: Previous map $\mathcal{M}_{k-1}$, features F_k^{multi}, state $\hat{x}_k$, covariance P_k, reliability field $U_{\text{pred}}^{(k)}$
Output: Updated dense map $\mathcal{M}_k$
Transform LiDAR and RGB-D points into the local map frame using $\hat{x}_k$
Associate points with active voxels and compute reliability weights $w_v = 1/(\epsilon + U_{\text{eff},v})$
Build kernel matrix using Eq. (S10) for active or inducing voxels
Estimate occupancy mean $\mu(x)$ and variance $\sigma^2(x)$ using GP regression or an inducing-point approximation
Update occupancy only when the reliability-weighted residual is within the selected gate
Fuse color or appearance only for reliable LiDAR–camera projections
return $\mathcal{M}_k$

Table S3: BDS and Uncertainty Prediction Parameters

Parameter	**Value used in submitted configuration**
LiDAR evaluation rate	10 Hz
Prediction horizon τ	0.1 s
Reverse diffusion steps T	100
Monte Carlo samples M	16
$\beta_{\min}$	1×10^{-4}
$\beta_{\max}$	0.02
Noise schedule	Cosine with clipping in Eq. (S3)
Time embedding dimension	128
Hidden dimension	512–1024 depending on module
Dropout during inference	0
Random seed	42

evaluation. M2UD is used only as an auxiliary benchmark for uncertainty analysis because its sensor configuration differs from the reported campus ground-robot platform. This separation avoids treating the M2UD uncertainty analysis as a direct hardware-identical trajectory benchmark.

S5.1. Metric Computation

LiDAR timestamps define the evaluation clock. RGB-D frames are associated with LiDAR frames using a fixed nearest-neighbor tolerance, and

Table S4: Training and Optimization Configuration

Parameter	**Value**
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-5}
Batch size	4
Gradient accumulation	8 steps when memory-limited
Gradient clipping	1.0
Mixed precision	Enabled for CUDA inference/training where stable
Loss terms	Synchronization, calibration, registration, mapping, diffusion regularization, and uncertainty regularization
Evaluation alignment	Ground-truth poses are interpolated to LiDAR evaluation time and rigidly aligned before ATE/RPE calculation

Table S5: DIKF Noise and Consistency Settings

Item	**Setting**
State	Rotation, position, velocity, accelerometer bias, gyroscope bias.
Process covariance	$Q_k = Q_{\text{base}} + \Sigma_\theta^Q(U_{\text{pred}}^{(k)})$.
Measurement covariance	$R_k^{\text{eff}} = R_{\text{base}} + \Sigma_\theta^R(\epsilon_{\theta,k}) + \Sigma_\theta(k)$.
Update rate	LiDAR-frame update at 10 Hz.
Consistency check	Normalized innovation squared gate with chi-square threshold selected by residual dimension.
Pose error metric	ATE and RMSE from aligned trajectory error; RPE from consecutive synchronized poses.

IMU samples are integrated between consecutive LiDAR frames. ATE is computed after rigid alignment to the ground-truth trajectory. RPE is computed between consecutive evaluated poses. The trajectory RMSE reported in the main article is computed from the aligned trajectory error over the synchronized campus evaluation frames used by the reported protocol.

S5.2. Reproducibility Checklist

For each reported run, the following items should be stored with the result files:

- sequence identifier and split name;

Table S6: POF Mapping Parameters

Parameter	**Value or rule**
Voxel resolution	0.1 m unless otherwise stated
Map update rate	LiDAR-frame update
RGB/depth/LiDAR kernel	Spatial RBF terms with learned signal and length-scale parameters
Diffusion kernel	Temporal reliability term controlled by predicted uncertainty
Observation weighting	Inverse relationship with U_{eff}
Color integration	Applied only for reliable LiDAR–camera projections
Unreliable updates	Down-weighted or gated before occupancy integration

Table S7: Campus Dataset Sequence Statistics

Sequence	**Len. (m)**	**Dur. (min)**	**Speed (m/s)**	**Scene**	**GT**
Seq-01 Corridor	245	6.2	0.66	Indoor	Total station
Seq-02 Lab	310	7.8	0.66	Indoor	Total station
Seq-03 Street	580	11.4	0.85	Outdoor	RTK-IMU
Total	**1135**	**25.4**	**0.74**	–	–

- random seed;
- sensor timestamp files and association tolerance;
- calibration prior and updated calibration file;
- evaluation script version;
- ground-truth source and alignment method;
- trained model checkpoint and configuration file;
- raw per-frame timing log used to compute runtime.

S6. Supplementary Figures

Figures S1 and S2 are included as supplementary qualitative material. They support interpretation of the PUT and trajectory replay but do not introduce a new benchmark protocol or replace the quantitative tables in the main article.

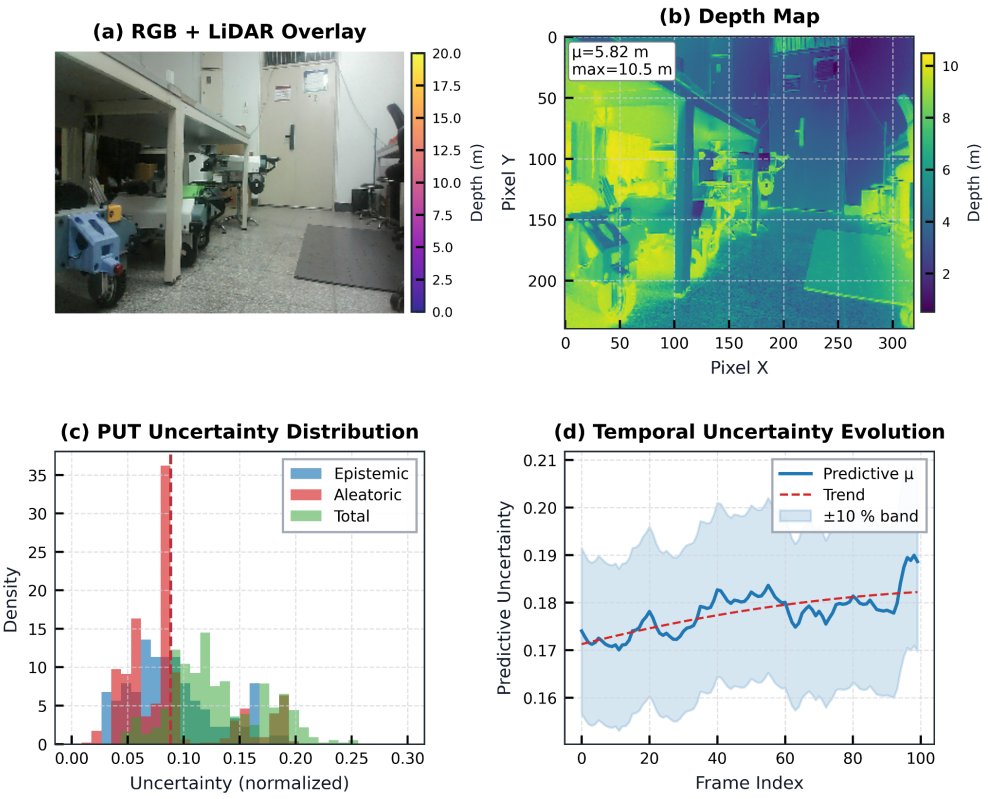

Figure S1: Supplementary PUT construction visualization. (a) RGB frame with LiDAR support overlay. (b) Depth map for the displayed frame, reporting the mean and maximum depth shown in the panel. (c) Distribution of epistemic, aleatoric, and total uncertainty for the displayed window. (d) Temporal evolution of predictive uncertainty over the plotted frame window. The figure clarifies how geometry, RGB-D support, and uncertainty channels are organized before synchronization and filtering.

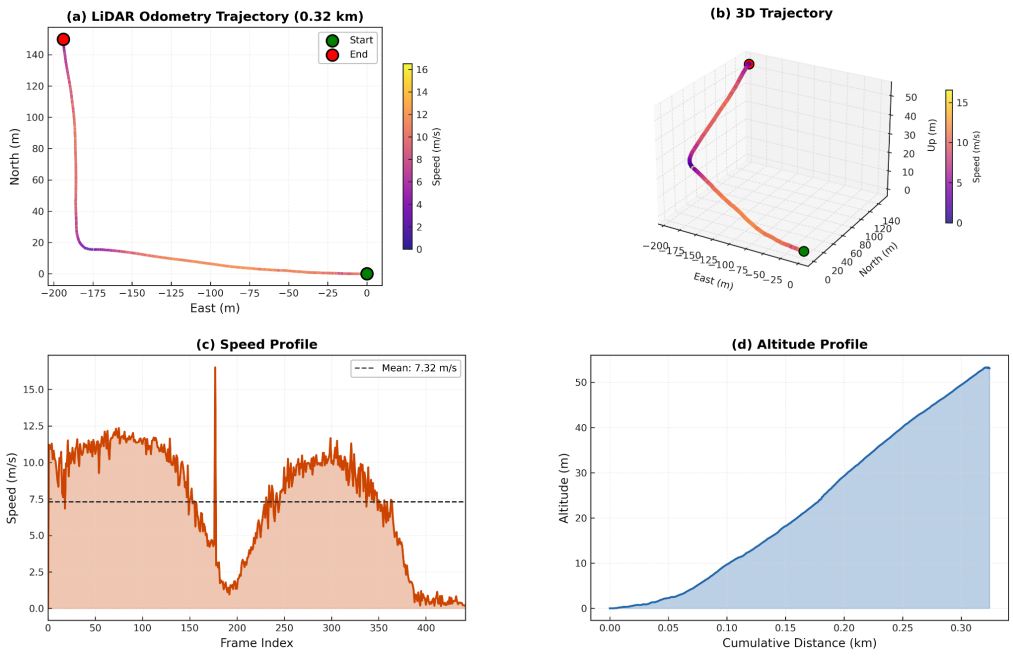

Figure S2: Supplementary trajectory inspection for a processed LiDAR odometry replay. (a) Plan-view trajectory with start and end markers; the plotted trajectory length is approximately 0.32 km, as indicated in the panel. (b) 3-D trajectory visualization. (c) Speed profile across replay frames. (d) Altitude profile with cumulative distance. This figure is used for qualitative coverage and consistency checking of the plotted replay and should not be interpreted as the full 1135 m campus dataset.

S7. Runtime and Resource Use

The main article reports an end-to-end runtime of 42 ms per LiDAR frame on the tested RTX 4060/Core i9 workstation, below the 100 ms LiDAR period. Table S8 gives a matching module-level runtime budget. If the implementation is re-profiled, this table should be regenerated directly from the per-frame timing log.

S8. Result-Consistency Notes for Submission

The following wording constraints were applied in this supplementary material to keep the document consistent with the main article and to avoid overstating the results:

Table S8: Runtime Budget per LiDAR Frame on the Tested Workstation

Module	**Time (ms)**	**Main processor**
Sensor association and PUT construction	4.2	CPU/GPU
BDS reliability prediction and denoising	9.0	GPU
DAMC calibration update	3.1	CPU
SE(3)-consistent feature extraction	6.5	GPU
DIKF propagation and update	5.2	CPU
POF dense mapping update	6.8	CPU/GPU
Map output, logging, and visualization buffer	3.4	CPU
Synchronization and overhead	3.8	CPU
Total	**42.0**	–

- The method is described as a reliability-aware LiDAR–RGB-D–IMU fusion system for dense local mapping and odometry, matching the main article title and scope.
- M2UD is described only as an auxiliary uncertainty-analysis benchmark, not as a direct trajectory-comparison dataset for the campus platform.
- Trajectory RMSE is described as being computed over the synchronized evaluation frames used by the reported protocol.
- The supplementary trajectory figure is explicitly labeled as an approximately 0.32 km replay visualization and not as the full 1135 m campus dataset.
- Runtime is described relative to the 10 Hz LiDAR update period and the tested workstation only.

S9. Conclusion

This supplementary material provides article-matched details for the reliability-aware LiDAR–RGB-D–IMU fusion system, including the sensor and computing platform, algorithmic pseudocode, parameter settings, dataset and evaluation notes, runtime budget, and the two added supplementary figures. The document is written to support reproducibility and figure interpretation without adding unsupported claims beyond the reported campus evaluation protocol.