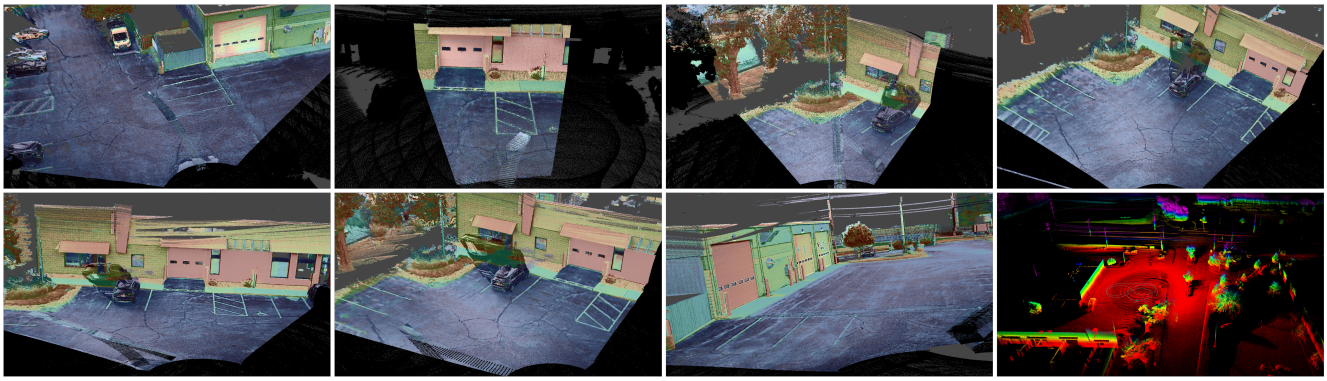
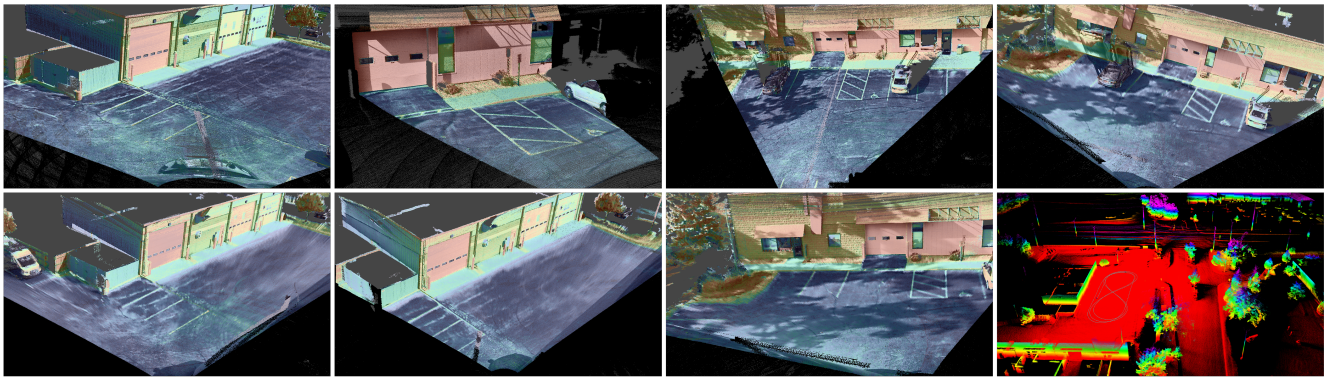
CATS-V2V: A Real-World Vehicle-to-Vehicle Cooperative Perception Dataset with Complex Adverse Traffic Scenarios

Supplementary Material



(a)



(b)

Figure 7. Calibration illustrations for our (a) red vehicle and (b) black vehicle. From top left to bottom right, figures illustrate the front wide-angle camera, front narrow-angle camera, rear camera, left front camera, left rear camera, right front camera, right rear camera, and INS calibration results, respectively.

In this Supplementary Material, we further introduce details about dataset implementation, dataset preprocessing of supporting tasks, metrics and evaluation of supporting tasks, and illustrations of proposed alignment methods.

A. Dataset implementation

A.1. Time synchronization

All sensors in our data collection process are hardware-synchronized to GNSS time. In addition to operating under the same clock, they are also triggered using the same PPS signals at integer seconds.

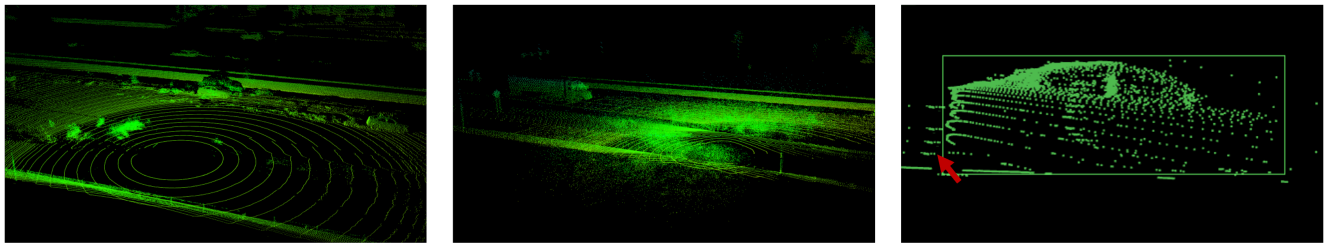
Each LiDAR operates in phase-lock mode at 0° azimuth. Therefore, at every rising edge of the PPS signal (i.e., each GNSS integer second), the LiDAR scan cycle is reset, and the azimuth counter returns to exactly 0°.

All cameras operate at 30 Hz and are triggered by a 33.333 ms pulse train derived from the same PPS origin. The first camera frame after each PPS aligns exactly with the GNSS integer second (as well as the corresponding LiDAR frame), while subsequent frames are triggered at fixed 33.333 ms intervals until the next PPS boundary.

The INS inherently operates under the GNSS clock and outputs 125 records per second. Within each second, the first record is emitted exactly at the integer second, with the remaining 124 records evenly spaced at 8 ms intervals.

A.2. Sensor calibration

Following Koide et al. [17], we conduct LiDAR-camera calibrations to achieve their precise extrinsic relationships for each LiDAR-camera pair across two vehicles. Each calibration converges when the normalized information distance (NID) change falls below 10^{-8}. To intuitively evaluate the calibration results, illustrations of camera image projecting



(a) (b) (c)

Figure 8. LiDAR frames affected by adverse weather. (a) Point clouds affected by raindrops, forming diffuse noise over mid-long distances. (b) Point clouds affected by snowflakes, forming clustered noise at close range. (c) In addition to noise, interfering water vapor behind vehicles will happen in rainy and snowy weather.

back into the point cloud frames are shown in Fig. 7.

Additionally, we follow Zhu et al. [43] and conduct LiDAR-IMU calibration through LiDAR odometry with inertial preintegration. To achieve sufficiently excited motion containing both rotational and translational components, we drive in different directions on a large and sloping ground. With these preparations, our calibration achieved a rotation accuracy of 0.1° and a translation accuracy of 2 cm. An intuitive illustration using LiDAR-inertial odometry is also shown in Fig. 7, in which the outstanding mapping quality represents the accurate calibration.

A.3. Data acquisition and anonymization

Our dataset was collected under diverse complex and adverse traffic scenarios, including weather such as clear, overcast, rainy, and snowy, lighting conditions such as direct sunlight or nighttime, as well as work zones and snow-covered roads, etc. Weathers have a direct impact on both LiDAR and cameras, while cameras are more sensitive to lighting conditions. The respectively affected camera images are shown in Fig. 2, while we also include the noise and interference of LiDAR point clouds under rainy and snowy weather conditions in Fig. 8.

To protect privacy, we apply an anonymization pipeline that masks personally identifiable information across all images. Faces and license plates are automatically detected and blurred through a pretrained instance segmentation model. Furthermore, to ensure spatiotemporal geographic security, each clip has relative timestamps spanning from 0.1 s to 30.1 s, instead of absolute timestamps. Similarly, precise INS positions are transformed into a local coordinate frame, where the first position of the red vehicle in record serves as the origin, and all trajectories are offset within the xy-plane.

A.4. Preprocess

We here introduce the details of motion compensation in our dataset preprocess.

A LiDAR scan is accumulated over a fixed time window (typically 100 ms for 10 Hz operation), during which the ego-vehicle undergoes non-negligible motion. Without compensation, it causes geometric distortions and misalignment in the point cloud. To obtain a consistent scan, we perform per-point motion compensation using each point's timestamp and high-precision poses from INS.

Let t_0 denote the start time of the scan, and t_i represents the timestamp of the i-th LiDAR point $\mathbf{p}_i(t_i)$ in the instantaneous LiDAR frame. The INS provides discrete LiDAR poses $\{\mathbf{x}(t_k), \mathbf{q}(t_k)\}$ at times $\{t_k\}$ through LiDAR-INS transformation. The continuous-time ego pose can then be linear and spherical linear interpolated by:

$$\mathbf{x}(t_i) = (1-\alpha)\,\mathbf{x}(t_a) + \alpha\,\mathbf{x}(t_b), \tag{2a}$$

$$\mathbf{q}(t_i) = \mathrm{SLERP}\left(\mathbf{q}(t_a),\, \mathbf{q}(t_b),\, \alpha\right), \tag{2b}$$

where (t_a, t_b) are the two closest INS timestamps such that $t_a \le t_i \le t_b$, and

$$\alpha = \frac{t_i - t_a}{t_b - t_a}. \tag{2c}$$

This yields the estimated continuous-time pose matrix:

$$T(t_i) = \begin{bmatrix} R(t_i) & \mathbf{x}(t_i) \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{2d}$$

$$R(t_i) = \mathbf{q}(t_i) \to \mathrm{SO}(3). \tag{2e}$$

Each LiDAR point is then transformed into the reference frame via:

$$\mathbf{p}_i(t_r) = T^{-1}(t_r)\,T(t_i)\,\mathbf{p}_i(t_i), \tag{3}$$

producing a geometrically consistent and deskewed point cloud at time t_r.

Intuitively, raw LiDAR scans exhibit a closed-ring structure because the azimuth sweeps return to 360°/0° at the end of each revolution. After motion compensation, however, the scan becomes an open structure, as shown in Fig. 9.

This opening directly reflects the physical motion of the ego vehicle and leads to a deskewed and aligned structure for further processing.

B. Dataset Processing of Supporting Tasks

As discussed in Sec. 4, we provide standardized data normalization pipelines for a wide range of tasks. Each task family requires specific combinations of modalities and coordinate representations. In this supplementary section, we detail a precise data format that allows researchers to directly integrate the data into a wide range of perception, spatial understanding, multi-modal learning, and cross-modal transfer systems, such as 3D detection, tracking, SLAM/odometry, 3D reconstruction, HD map generation, joint compression, depth estimation, view synthesis, and scene transfer, without requiring any additional preprocessing(the tasks summarized in Table 4 of the main paper).

B.1. 3D Detection and Tracking

For 3D detection and multi-frame 3D tracking, we unify all annotations into the **nuScenes** data representation, which provides a structured and comprehensive description of multi-sensor data, ego-motion, and object trajectories. The hierarchy after conversion follows:

- **scene**: a continuous driving sequence (∼30 seconds);
- **sample**: keyframes at 2 Hz, each representing a synchronized multi-sensor snapshot;
- **sample_data**: sensor-specific data (LiDAR sweeps, images, radar returns, etc.);
- **sample_annotation**: 3D bounding boxes for objects within each sample;
- **instance**: the full lifespan of an object within a scene;
- **category / attribute / visibility**: auxiliary descriptors;
- **ego_pose**: the 6-DoF pose of the ego vehicle;
- **calibrated_sensor / sensor**: extrinsic and intrinsic calibration parameters;
- **map / log**: region-level map metadata and sequence logs.

All synchronization and keyframe selection use LiDAR timestamps as the temporal reference, with keyframes uniformly sampled at 2 Hz following the nuScenes protocol.

The ego pose for each nuScenes-style sample is computed using the same continuous-time pose interpolation strategy described in Appendix A.4. Specifically, INS measurements provide discrete 6-DoF poses, which are converted into continuous-time ego poses at arbitrary timestamps via linear interpolation for translation and spherical linear interpolation (SLERP) for rotation. This unified interpolation model ensures that the ego poses used in the nuScenes-format conversion are temporally consistent with the motion compensation procedure and all other preprocessing steps in CATS-V2V.

B.2. Map Generation

For HD map generation, we follow the **nuScenes** map and log representation. CATS-V2V provides structured map-related metadata that supports BEV map construction, static-scene understanding, and multi-pass aggregation. Specifically, we include:

- **Rasterized BEV layers**: height maps, occupancy grids, and intensity maps generated from deskewed LiDAR sweeps.
- **Static scene elements**: road boundaries, lane dividers, and drivable regions extracted from multi-pass fused point clouds.
- **Global coordinate frames**: GPS/INS-based world coordinates, ensuring consistent alignment across all passes by using the same ego-pose interpolation method described in Appendix A.4.
- **Log metadata**: region identifiers, timestamps, and scene information analogous to the nuScenes `log` entries.

B.3. SLAM and odometry

For SLAM and odometry research, we provide scripts to combine all video frames, raw LiDAR point clouds, and raw IMU measurements into a single ROS2 bag for each vehicle, enabling direct loading into existing SLAM frameworks without additional data conversion.

In addition, we provide scripts for convenient conversion between ROS1 and ROS2 bag formats.

B.4. 3D Reconstruction

For 3D reconstruction tasks, we convert each sequence into an **ETH3D-style format**, which is commonly used for multi-view stereo and dense reconstruction benchmarks. Each reconstruction package contains:

- **Undistorted multi-view images**: photometrically consistent RGB frames ready for multi-view processing.
- **Camera intrinsics and extrinsics**: stored in an ETH3D-style calibration file for direct compatibility with MVS and neural reconstruction pipelines.
- **Ego poses**: obtained by continuous-time interpolation of INS measurements (as described in Appendix A.4), ensuring globally consistent camera trajectories.
- **Deskewed and globally aligned LiDAR point clouds**: used as geometric priors for reconstruction and depth reasoning.
- **Approximate depth maps**: generated by first reconstructing a dense LiDAR-inertial map using **Fast-LIO2**, and then projecting the global map into each camera frame using the interpolated ego poses, the INS–LiDAR and LiDAR–camera extrinsics, and the camera intrinsics. These depth maps provide useful supervision for learning-based reconstruction while not being considered perfect ground truth.For each LiDAR point $\mathbf{P}^L \in \mathbb{R}^3$, we

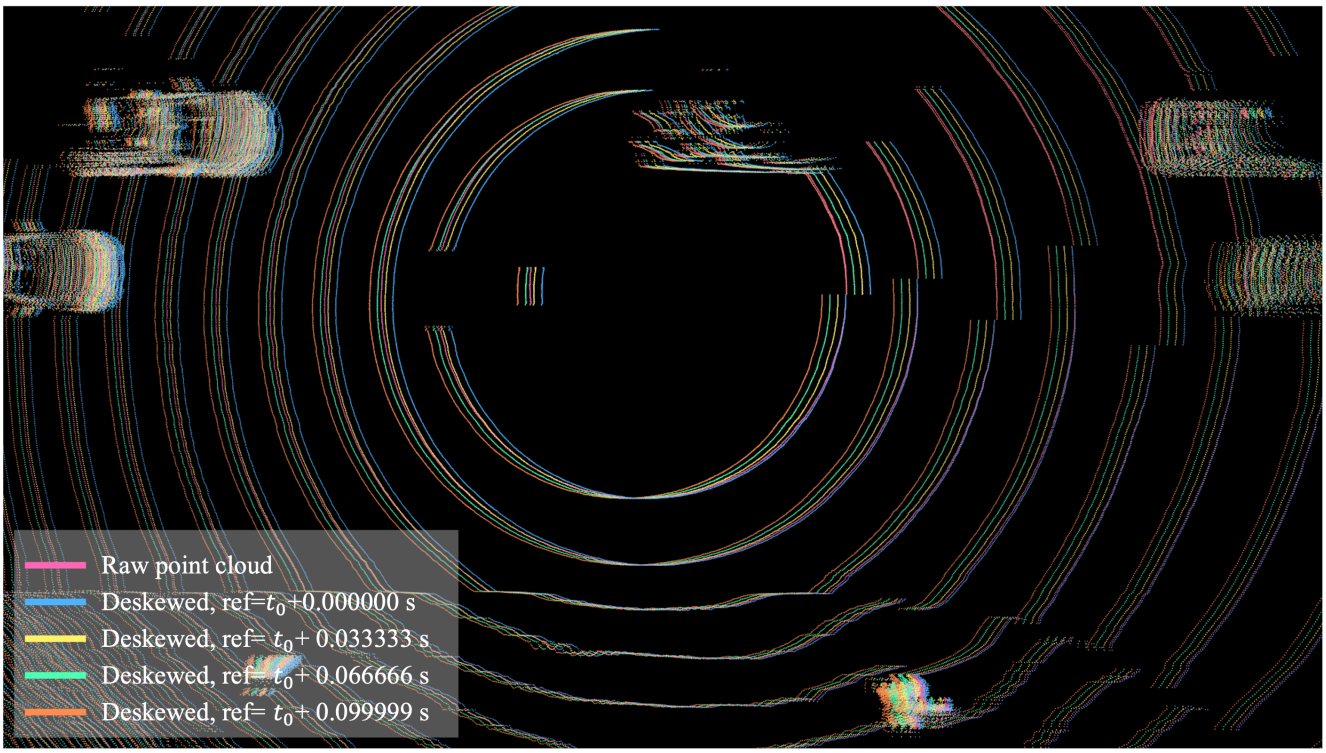


Figure 9. Illustration of motion compensation of a single frame to different time.

transform it into the camera frame via

$$\mathbf{P}^C(t) = T_C^L \, T_L^W(t) \begin{bmatrix} \mathbf{P}^L \\ 1 \end{bmatrix}, \tag{4}$$

and project it onto the image plane using the camera intrinsics

$$\mathbf{u} = \pi(\mathbf{P}^C) = \begin{bmatrix} f_x \frac{X^C}{Z^C} + c_x \\ f_y \frac{Y^C}{Z^C} + c_y \end{bmatrix}, \qquad D = Z^C. \tag{5}$$

This ETH3D-style representation enables direct integration with classical and modern 3D reconstruction pipelines without additional data conversion. But if fine-grained reconstruction is desired for only a specific region within a scene, users need further subset and filter the relevant point clouds to meet the requirements of their task.

B.5. Scene Transfer

For scene transfer and domain adaptation tasks, we provide filtered sets of frames captured from identical scenes under different environmental conditions. To automatically construct such cross-condition correspondences, we develop a simple yet effective frame filtering and matching strategy.

For each scene, the *overcast* condition is treated as the reference domain due to its clear and stable appearance. We first compute edge maps for all overcast frames and uniformly sample them at 1 Hz to reduce redundancy. Then, for each sampled reference frame, we identify the most structurally similar frame under another condition by matching their edge responses.

Formally, given the edge map E_r of a reference frame and the edge map E_t of a candidate frame under a different condition, we compute a normalized similarity score:

$$S(E_r, E_t) = \frac{\langle E_r, \, E_t \rangle}{\|E_r\|_2 \, \|E_t\|_2}, \tag{6a}$$

and select the frame with the highest similarity

$$t^* = \arg\max_t \, S(E_r, E_t). \tag{6b}$$

This procedure produces cross-condition frame pairs that are spatially aligned to the same physical locations. Only image data are provided for this task, and each matched set is further manually verified to ensure alignment accuracy and reliability for appearance transfer and cross-domain correspondence learning.

B.6. Other Tasks

For all remaining tasks supported by CATS-V2V, the required data normalization and preprocessing procedures follow directly from the formats and components already described in the preceding subsections. Since these tasks rely on the same standardized modalities, coordinate frames, and pose interpolation strategies, we omit additional elaboration here.

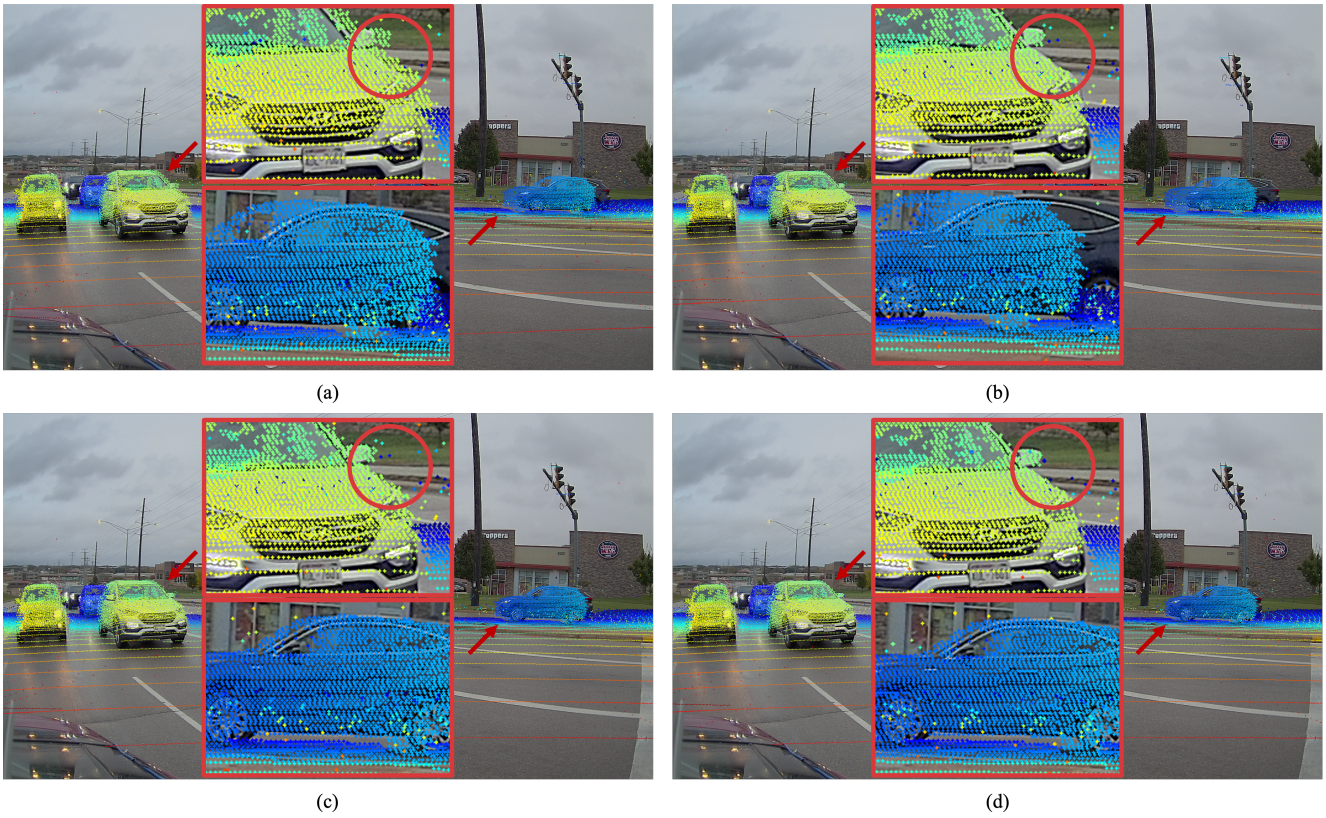


Figure 10. Comparison of alignment. (a) Stamp-based alignment. (b) Deskewing removes distortion for static vehicles but worsens alignment for moving ones. (c) Frame-matching aligns dynamic vehicles but leaves static distortions. (d) Frame-alignment achieves both static deskewing and dynamic consistency.

C. Metrics and Evaluation of Supporting Tasks

C.1. Object Detection

The evaluation of object detection typically follows standard metrics such as mean Average Precision (mAP) under category-specific IoU thresholds. For 3D detection, nuScenes-style metrics such as mAP and NDS can be applied directly.

To support model development and fair comparison, we provide official *train/val/test* splits at the scene level. Each split preserves geographical diversity, weather variations, and multi-pass structure to encourage generalization.

C.2. Object Tracking

Object tracking in CATS-V2V is defined on top of the detection annotations. For 3D tracking, annotations follow the nuScenes `instance` structure, where each object is assigned a unique identifier throughout an entire scene.

Tracking splits are aligned with the detection splits, ensuring consistent scene partitioning and preventing temporal leakage across sets.

C.3. SLAM

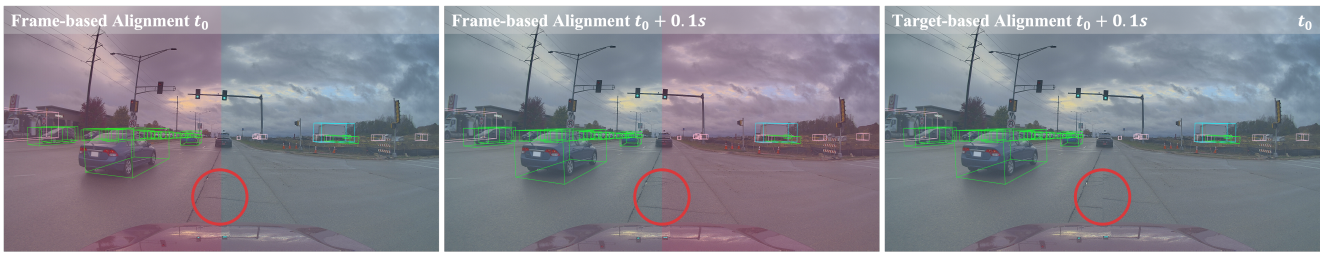
CATS-V2V provides high-precision GNSS/INS, multi-view RGB frames, and dual 128-beam LiDAR sweeps, enabling both LiDAR-based and visual-inertial SLAM.

Standard SLAM evaluation metrics (e.g., Absolute Trajectory Error and map completeness) can be applied. Additionally, recently proposed metrics such as AWD and SCS are fully supported with reference implementations included in our codebase.

For benchmarking, we provide several scenario-level sequences with ground-truth trajectories. Multi-pass data are grouped so researchers can study long-term localization, environmental changes, loop closure, or domain-shift robustness (e.g., weather, construction zones, seasonal variation).

C.4. HD Map Generation

Evaluation typically involves geometric accuracy, coverage completeness, and consistency across passes. We do not enforce a specific metric; instead, we provide clean multi-pass ground-truth maps and recommend using standard mapping metrics or task-specific benchmarks depending on the algorithmic setting.



(a) (b) (c)

Figure 11. A comparison between target-based alignment with frame-based alignment. (a) An image captured at the beginning of the LiDAR scan t_0. (b) An image captured at the end of the LiDAR scan $t_0 + 0.1$ s. (c) The left portion image corresponds to the end of the LiDAR scan $t_0 + 0.1$ s, while the right portion image corresponds to the beginning t_0, allowing targets to keep aligned on both portions.

C.5. Joint Compression

Since compression methods typically target *lossless* or *near-lossless* reconstruction, evaluation commonly relies on metrics such as compression ratio, bit-per-pixel/point, and reconstruction fidelity (e.g., PSNR/SSIM for RGB, Chamfer distance for LiDAR).

To facilitate fair evaluation, all RGB images in CATS-V2V are stored in `.png` format rather than JPEG. This prevents any pre-existing lossy compression artifacts from influencing algorithmic comparisons or biasing cross-modal reconstruction performance. LiDAR data are stored in raw point cloud or range-image tensors, preserving the original dynamic range and sensor characteristics.

Scene-level *train/val/test* splits follow the same partitioning strategy used in perception tasks, ensuring consistent cross-task benchmarking.

C.6. Domain Adaptation

Domain adaptation tasks in CATS-V2V involve shifts across weather, illumination, seasonal conditions, or multi-pass temporal variations. The diversity of domains enables evaluation across several downstream goals, including:

- **Cross-domain object detection/tracking**: evaluated using mAP, AMOTA, and other task-specific metrics introduced earlier;
- **Scene transfer and image-to-image translation**: evaluated using image-similarity metrics such as PSNR, SSIM, LPIPS, or perceptual FID-style measures;
- **Cross-modal or cross-view adaptation**: evaluated through reconstruction accuracy or geometric consistency metrics.

Given the complexity of domain shifts, we recommend conducting evaluation on a *per-scene* basis: each scene is treated as an independent domain, and models are assessed on their ability to generalize across scenes under different pass conditions (e.g., day $\rightarrow$ night, dry $\rightarrow$ snow, or before $\rightarrow$ after construction).

Per-Scene Domain Evaluation Let $\mathcal{S} = \{S_1, S_2, \ldots, S_K\}$ denote the set of scenes, each corresponding to a distinct environmental or temporal domain. A model is trained on a subset of scenes $\mathcal{S}_{\text{train}}$ and evaluated on one or more unseen scenes $\mathcal{S}_{\text{test}}$ to measure cross-domain transfer performance.

Let $\mathcal{D}(S_j)$ denote the evaluation data from scene S_j. Using object detection as an example, the per-scene mAP is computed as:

$$\text{mAP}(S_j) = \frac{1}{C} \sum_{c=1}^{C} \text{AP}_c\big(\mathcal{D}(S_j)\big), \tag{7a}$$

where C is the number of categories and AP_c is the average precision for the c-th category on scene S_j.

A final cross-domain score is obtained by averaging over all test scenes:

$$\text{mAP}_{\text{domain}} = \frac{1}{|\mathcal{S}_{\text{test}}|} \sum_{S_j \in \mathcal{S}_{\text{test}}} \text{mAP}(S_j). \tag{7b}$$

D. Alignment

A comprehensive comparison from simple stamp-based alignment to frame-based alignment is shown in Fig. 10, where we project the corresponding point clouds onto the images for a better illustration of deskewing.

An intuitive comparison between target-based alignment and frame-based alignment is further shown in Fig. 11, where we take the wide-angle front-view camera's image as a typical example. The right portion of the image corresponds to the beginning of the LiDAR scan (t_0), whereas the left portion corresponds to the end ($t_0 + 0.1$ s).

When conducting target-based alignment, as shown in Fig. 11, targets are aligned to two different images captured at different timestamps accordingly, allowing bounding boxes and objects on both left and right portions to appear spatial-temporally consistent.