CALIBBENCH: DIVERSE CROSS-MODAL CALIBRATION BENCHMARK

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-modal sensor fusion is integral to modern autonomous systems for perception and interaction with their environments. The safe and successful deployment of these systems depends on ensuring that the distinct sensor modalities are aligned, necessitating cross-modal calibration to identify the extrinsic parameters among the sensors. Current methods seek to address the need for cross-modal calibration in various ways, including target-based and target-less calibration using data-driven approaches. However, existing methods have been evaluated on homogeneous or closed-source datasets, which makes it difficult to fairly compare, transfer, and validate in different settings. In this paper, we present CALIBBENCH, a diverse cross-modal (camera and LiDAR) calibration dataset, and use it to benchmark existing methods on heterogeneous extrinsics. Through our benchmark, we demonstrate the state-of-the-art in cross-modal calibration, the challenges that diverse extrinsics pose, and the struggle in improving performance during retraining. The benchmark results reveal research gaps in accuracy, robustness, and domain transferability, motivating cross-modal calibration research for arbitrary sensor configurations in real-world deployments.

1 INTRODUCTION

Modern autonomous systems leverage a variety of sensors to perceive, navigate, and interact with their environment. The availability of these sensors and associated multi-modal fusion techniques enables various applications, ranging from robots navigating in crowded indoor environments (Sathyamoorthy et al., 2020) to safely deploying autonomous vehicles on roads (Liu et al., 2023; Li et al., 2022). Although the performance of cross-modal solutions on public benchmarks has improved in recent years, deploying these models in the real world with an arbitrary sensor setup often requires an extensive recalibration process (Teng et al., 2023). Further, it has been shown that an extrinsic calibration error of even a few degrees in rotation or a few centimeters in translation can compound over a distance (*e.g.,* rendering a 20 cm displacement at 5 meters distance) (Huang & Grizzle, 2020), yielding misalignment and degraded fusion performance.

Existing public datasets (*e.g.,* KITTI (Geiger et al., 2013) are not designed for such calibration tasks. The sensor placement is typically fixed across different traces/sequences. There is little to no variation in the extrinsics; hence, models trained on these datasets can easily overfit and fail to generalize. Although attempts have been made to train models on custom datasets (Yang et al., 2024; Sen et al., 2023), these datasets are closed-source, making fair comparisons with other methods difficult. Furthermore, there are scenarios in which dynamic calibration is required. For example, in autonomous trucks (Zhu, 2026), sensors are mounted on both the cab and trailer, which are not rigidly joined and have no kinematic feedback. Consequently, the extrinsics between the sensors would change over time. Other systems with dynamic components, such as multi-arm and reconfigurable robots, can also leverage dynamic calibration.

This work aims to provide an open-source, diverse, cross-modal calibration dataset that addresses the challenges of limited extrinsic variation and fixed sensor placement. In the developed framework (shown in Fig. 1), we use a bimanual robotic system to precisely place sensors at a wide range of 3D poses in a targetless setting. Pose information of the camera and LiDAR sensors mounted on the arms is tracked using an offboard motion capture (MoCap) camera system, allowing the extrinsic transformations between these sensors to be calculated precisely.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We collect both ***discrete*** and ***continuous*** camera and LiDAR traces with both ***static*** and ***dynamic*** extrinsics. The traces are organized by increasing levels of difficulty, offering diverse and challenging benchmark and training datasets for both data-driven and non-data-driven approaches.

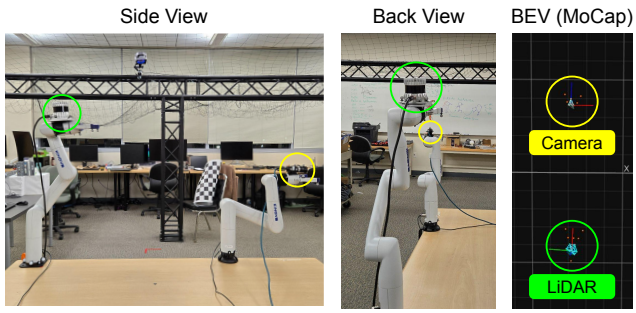


Figure 1: CALIBBENCH is a diverse multi-modal calibration dataset. We utilize a motion capture system to track the LiDAR and camera sensors, both mounted on two bimanual robotic manipulators. The arms move along different trajectories while the poses of the sensors are tracked. This enables the collection of diverse ground-truth extrinsics between the sensors, a feature not present in current benchmarks for multi-modal calibration.

In summary, we deploy a bimanual robotic platform to collect a large-scale cross-modal calibration dataset. Using this platform, we develop a cross-modal (camera and LiDAR) calibration dataset with diverse ground-truth extrinsics. The dataset is categorized into discrete or continuous sequences, with continuous sequences ranging up to 30 seconds, with varying levels of difficulty and static or dynamic extrinsics. In total, CALIBBENCH consists of **38,442 frames** across **two indoor scenes** and **four outdoor scenes**. We then train and conduct thorough benchmark testing of state-of-the-art cross-modal calibration methods on our dataset to assess their efficacy on CALIBBENCH. We anticipate that the work herein will serve the community by allowing researchers and practitioners to further refine and improve the frontier of cross-modal calibration.

2 RELATED WORK

Target-based vs. Target-less Methods. Target-based calibration methods rely on a planar calibration pattern, *e.g.,* a checkerboard or an AprilTag grid, as a reference for extrinsic calibration. Earlier works (Zhang & Pless, 2004; Huang & Grizzle, 2020) found that LiDAR scans on the planar pattern can be used to register constraints with the estimated pattern on the camera image plane for reliable extrinsic calibration. Yan *et al.* (Yan et al., 2023) provides a method to jointly calibrate camera intrinsics and LiDAR to camera extrinsics using a special type of target with checkerboards and conic sections. However, the reliance on targets makes these methods tedious and laborious for frequent calibrations required when deploying robotic systems. In addition to target-based calibration methods, there exist target-less calibration methods that do not rely on a planar calibration pattern. For example, Ishikawa *et al.* (Ishikawa et al., 2018) propose the use of motion to facilitate extrinsic calibration in a target-less setting. Pandey *et al.* (Pandey et al., 2012) take a probabilistic approach to maximize information for cross-modal calibration.

Hybrid vs. Data-driven Methods. With the shift in incorporating neural networks into many tasks, target-less calibration methods have also given rise to learning-based methods, which can be either hybrid-based or completely data-driven. Hybrid-based models (Koide et al., 2023; Luo et al., 2024; Huang et al., 2024) use vision models, *e.g.,* SuperGlue (Sarlin et al., 2020) and SegmentAnything (Kirillov et al., 2023; Zhang et al., 2023), to extract features and find a dense correspondence between image features and LiDAR points before using classical optimization methods.

Data-driven methods are trained on large amounts of data and learn the correspondences between LiDAR and image modalities to regress or refine extrinsic parameters in the targetless setting. Early CNN-based approaches (Wu et al., 2021; Schneider et al., 2017; Shi et al., 2020) estimate extrinsics directly from LiDAR-derived depth representations and RGB images and explored incorporating temporal modeling through LSTMs (Hochreiter & Schmidhuber, 1997). In contrast, iterative refinement approaches (Lv et al., 2021; Iyer et al., 2018; Ou et al., 2026; 2025) optimize calibration from an initial estimate using geometric constraints and spatial transformer formulations. More recent methods improve cross-modal correspondence learning through dense correlation reasoning and structured scene representations, *e.g.,* RAFT-style correlation volumes (Lee & Chen, 2024) and bird-eye-view (BEV) representations (Yuan et al., 2025; D'Addeo et al., 2026). Neural radiance field (NeRF)-based methods (Mildenhall et al., 2021; Yang et al., 2024; Herau et al., 2024) have

also emerged as a new paradigm for cross-modal calibration, where the reconstructed scene acts as a pseudo-target for cross-modal alignment. Here, we focus on a benchmark for data-driven methods.

Datasets for Cross-modal Calibration. Autonomous driving datasets are commonly used as evaluation benchmarks in the cross-modal calibration literature (Lv et al., 2021; Lee & Chen, 2024; Herau et al., 2024; Yang et al., 2024) because of the availability of cross-modal sensor streams and their precise calibration parameters. Among those datasets, the most frequently used are KITTI (Geiger et al., 2013), NuScenes (Caesar et al., 2020), and Pandaset (Xiao et al., 2021). They contain LiDAR-camera extrinsics, as well as thousands of frames collected across multiple days and in different cities; these features make them particularly useful for training data-driven methods. However, the extrinsics of these datasets are homogeneous because they are primarily aimed at autonomous driving rather than general-purpose calibration; therefore, the positions of the sensors are fixed. Other works (Koide et al., 2023; Sen et al., 2023; Yang et al., 2024) collected their own datasets to evaluate their distinct sensor configurations, but they are also homogeneous in the extrinsics that they provide. Furthermore, most of these datasets are not open-source, making comparisons with competing and derivative methods difficult.

3 CALIBBENCH METHODOLOGY

In this section, we describe the CALIBBENCH data collection methodology, including an overview of the system setup, MoCap ground-truth acquisition methods, and data validation. Details regarding sensor positioning and synchronous data collection are in Appendix Section B.

System Setup Overview. The CALIBBENCH system comprises three components: an extrinsic ground truth module (MoCap), a sensor pose manipulation module (Robot Arms), and a synchronous data collection and validation module. Fig. 2 shows an overview of the setup and the corresponding recorded data. The MoCap system first returns the poses of both the sensors and the end-effectors of the arms. The arms move the sensors/ to provide different views of the environment, while the collection module records both the sensor streams and the arm end-effector poses via ROS2. Collected data are synchronized with MoCap recordings to match the MoCap and arm poses. In all, the following data are produced for each recorded frame: *timestamp, camera and LiDAR poses, end-effector poses, sensor extrinsics, images, point cloud data, and camera intrinsics*.

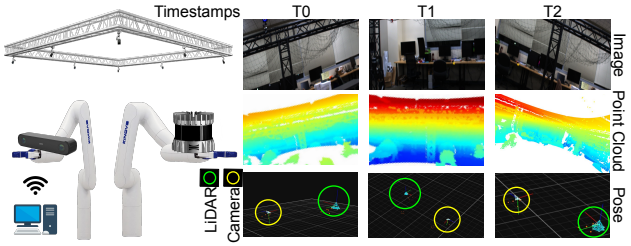


Figure 2: CALIBBENCH Overview. While the sensors collect our multimodal data, the MoCap system records and sends the tracked rigid bodies' poses. The released data format contains synchronized images and point clouds with timestamps, poses, and ground-truth extrinsics.

Accurate Extrinsics. The availability of accurate ground truth extrinsics is critical to the success of a diverse CALIBBENCH. However, when the sensors are moving, it becomes nontrivial to capture their extrinsics precisely. We use a wide-area (25' x 25' x 7') 8-camera Optitrack motion capture system to obtain CALIBBENCH's ground truth (mm-level accuracy) poses and calculate extrinsics thereafter. In the MoCap area, we set up two Kinova Gen3 Lite robot arms and mount a Lucid Triton 2.3 MP camera and an Ouster OS1-128 LiDAR right behind the last joint. Markers are placed asymmetrically on the sensors and end-effectors of the arms to help resolve ambiguities and maintain the orientation tracking of the rigid bodies. The MoCap system then tracks the poses of the sensors and the robot end-effectors. Given the poses of the rigid bodies of the sensors, we can calculate the extrinsic from the LiDAR to camera rigid bodies as $\widehat{\mathcal{T}}_{\mathcal{L}}^{\mathcal{C}}$ as an initial guess. The marker placement introduces human error that causes the defined rigid bodies to not capture the true optical center of the sensors. This is remedied with eye-hand calibration (Remy et al., 1997) on both sensors to complete the rigid-body transformation between sensors. Eye-hand calibration is formulated as solving for X in $AX = XB$. A is the relative motion between the sensor rigid bodies' poses defined by the MoCap markers, $\mathcal{T}_{markers}^{mocap}$, and B is the relative motion between a fiducial target to sensor transform, $\mathcal{T}_{target}^{sensor}$. The latter set of transforms can be calculated through the Perspective-n-Point algorithm using fiducial targets, such as Charuco boards and April tags. We define $\mathcal{T}_{\mathcal{L}}$ and $\mathcal{T}_{\mathcal{C}}$ as

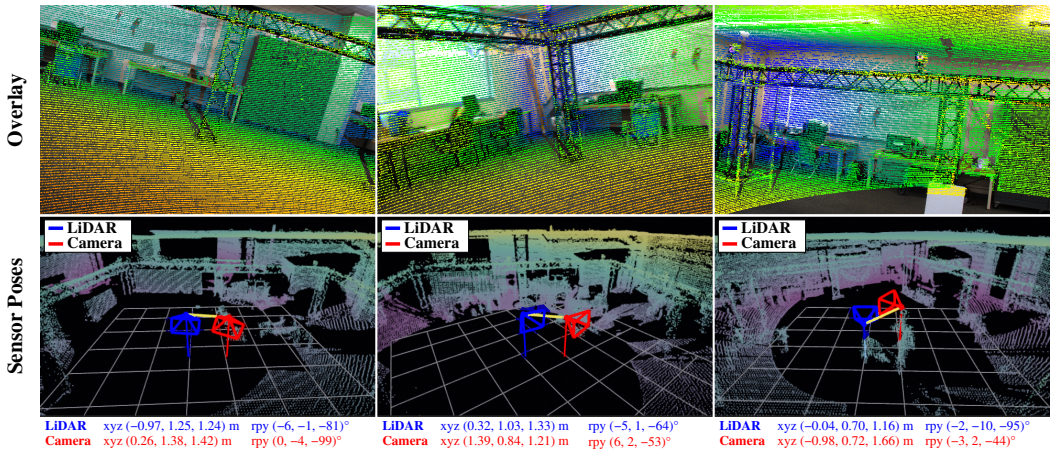


Figure 3: LiDAR-camera overlay and their corresponding extrinsic poses.

the eye-hand transforms for the LiDAR and camera sensors, respectively. The final ground-truth extrinsic can be computed as $\mathcal{T}_{\mathcal{L}}^{\mathcal{C}} = \mathcal{T}_{\mathcal{C}} \times \widehat{\mathcal{T}_{\mathcal{L}}^{\mathcal{C}}} \times \mathcal{T}_{\mathcal{L}}$. We follow the established procedure for camera eye-hand calibration and provide details for LiDAR eye-hand calibration in Appendix Section C.

Data Validation. For this dataset to be valuable to the community, we must ensure that the ground-truth transformations between the sensors are accurate. This can be confirmed through reprojection analysis using the same fiducial target used in the eye-hand calibration. For each calibration sample i, we have the sensor's pose in MoCap $\mathcal{T}_{markers_i}^{mocap}$ and the per-sample target pose in the sensor frame $\mathcal{T}_{target_i}^{sensor}$. With the eye-hand transform X, we obtain a consistent estimate of the target's pose in MoCap as $\mathcal{T}_{target}^{mocap} = \frac{1}{N}\sum_i \mathcal{T}_{markers_i}^{mocap} \cdot X \cdot \mathcal{T}_{target_i}^{sensor}$, and predict each sample's target pose in the sensor frame as $\mathcal{T}_{target}^{sensor} = X^{-1} \cdot (\mathcal{T}_{markers_i}^{mocap})^{-1} \cdot \mathcal{T}_{target}^{mocap}$. The target's 3D corners are projected into the sensor's image via the predicted $\mathcal{T}_{target}^{sensor}$, and the reprojection error is computed as the L2 norm between these reprojected corners and the corners observed by the sensor. For the camera, corners are detected via AprilTag/Charuco board; for the LiDAR, corners are extracted from the point cloud via a joint Gauss-Newton fit. In our eye-hand calibration, we achieve mean pixel reprojection errors of 2.77 and 1.80 for LiDAR and camera, respectively. We note that the image sizes for the LiDAR image and camera image are 1280 x 530 and 1920 x 1200 pixels, respectively, and during eye-hand calibration and evaluation, the fiducial target encompasses the vertical field of view of the sensors. Fig. 3 visualizes a LiDAR point cloud overlaid on top of a camera image after applying the eye-hand transforms on both sensors. The point cloud corresponding to the metal truss structure, desks, chairs, and printers overlay well on the objects in the scene. Fig. 4 further confirms our extrinsic quality by achieving lidar-camera overlays identical to those produced by target-based calibration methods, which are known to be accurate. Additional details on extrinsic validation are provided in Section 5, and more details on reprojection analysis is in Appendix Section D. By achieving a low reprojection error and observing the LiDAR-camera overlays, the calculated extrinsics are validated as ground-truth extrinsics.

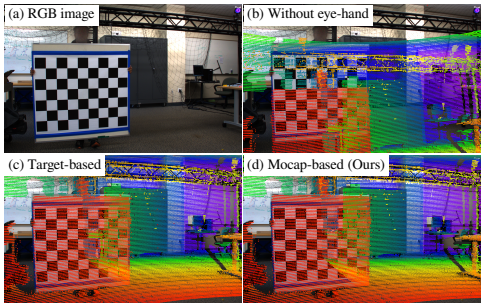


Figure 4: Our eye-hand derived extrinsic closely matches the one calculated by target-based calibration.

4 CALIBBENCH DATASET

In this section, we describe the traces of CALIBBENCH. We group the dataset into four groups, identified by two factors, ***discrete*** vs. ***continuous*** traces, with ***fixed*** vs. ***dynamic*** extrinsics, which are the primary differences between existing datasets used for cross-modal calibration. Each continuous trace spans roughly 30 seconds, providing temporal variation for motion-based calibration. With these properties, we intend to address the limitations of previous datasets, as shown in Table 1.



Figure 5: Sample sequence of the *Continuous Trace with Static Extrinsic* ***(CS)***.

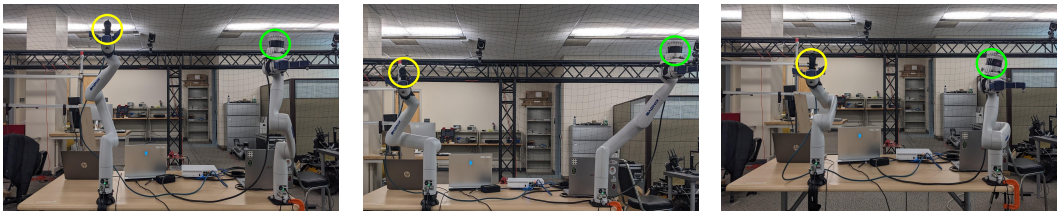


Figure 6: Three separate sequences of the *Discrete Traces with Dynamic Extrinsics* ***(DD)***.

In particular, the targetless setting enables research for calibration where specialized equipment is not available or when self-calibration in the wild is required. We also describe the trajectories of increasing difficulty levels for calibration purposes.

Table 1: CALIBBENCH provides dynamic and unique sensor placements, while existing datasets have only static sensor extrinsics.

Dataset	#Seq.	Discrete	Continuous	Static	Dynamic
KITTI (Geiger et al., 2013)	156	✓	✓	✓	
NuScenes (Caesar et al., 2020)	1000		✓	✓	
PandaSet (Xiao et al., 2021)	100+		✓	✓	
CALIBBENCH (Ours)	170	✓	✓	✓	✓

Continuous Traces, Static Extrinsics (CS). Continuous static (CS) traces are the easiest setting that is available in public autonomous driving benchmarks. Herein, sensors are mounted on one rigid body (*e.g.,* a self-driving car or a robot), and that body moves continuously during sensor data recording. In CALIBBENCH, both robot arms remain stationary, keeping the extrinsic stable and are placed on a movable base (as shown in Fig. 5). Recording such data is motivated by the challenge of calculating the extrinsics while moving. Also, some calibration methods (Koide et al., 2023; Lee & Chen, 2024) can leverage temporal frames to improve the final extrinsic prediction.

Discrete Traces, Static Extrinsics (DS). Discrete static (DS) traces are a similar, simple setting as the continuous version above, except that there is no temporal history to refer to. All calibrations must be performed using a single frame of sensor data. The data of these traces are a discrete subset of the continuous version.

Discrete Traces, Dynamic Extrinsics (DD). Discrete dynamic (DD) traces are a slightly more challenging setting. In these traces, each LiDAR and camera pair is taken from different extrinsics. Calibration methods are still only allowed to use just this pair with no other temporal information. In collecting these traces, we start with simple extrinsic changes, *e.g.,* move one arm along one axis while keeping the other arm static, and then move on to complicated maneuvers, *e.g.,* moving one arm arbitrarily, and moving two arms together. The objective is to move either robot arm in a way that captures various ground-truth extrinsics. We repeat this process multiple times until sufficient sequences are collected. Fig. 6 illustrates a few examples of different poses used in this category.

Continuous Traces, Dynamic Extrinsics (CD). In this setting, continuous dynamic (CD) traces are continuously recorded while the sensors and the arms are moving. This is a relatively challenging case, yet it is applicable to a few robot applications where sensors are mounted on the movable parts of each robot. The task is to continuously compute the relative transformation between the two sensors in real time so that the extrinsics of each frame in the continuous trace are accurate. In this case, the temporal history can be used in those models (Koide et al., 2023) that take *sequences* of data as input.

Similar to the continuous traces with static extrinsics, we first move the arms in the single axis x-, y-, and z- directions. The arm with the LiDAR sensor mounted remains stationary during the recording but is moved in preparation for the next sequence. We also record the transformation between the end-effector and the arm's base frame. This helps us track and correlate the trajectories in MoCap and what the robot arm reports, so we can align them by determining the offset between trajectories.

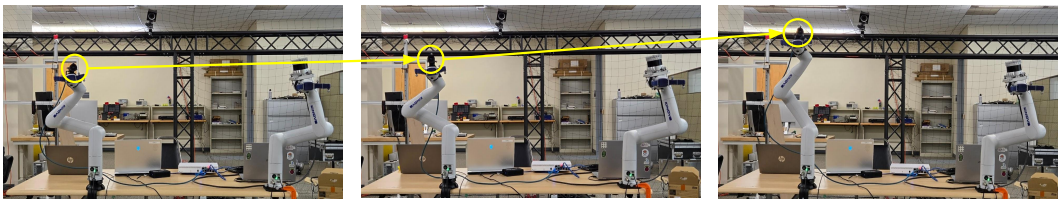


Figure 7: Sample sequence of the *Continuous Traces with Dynamic Extrinsics* ***(CD)***.

Thus, we can determine how to synchronize the ground-truth extrinsics from MoCap with the image and point cloud data recorded from ROS. The traces of the discrete version above are a discrete subset of this category. Sample sequences from this case are depicted in Fig. 7.

Diversity and Scale. Table 2 provides the breakdown of CALIBBENCH's diversity and scale. We collected 170 trajectories in two indoor and four outdoor environments. In total, there are 38,442 pairs of image and point cloud data with corresponding extrinsics. Trajectories using dynamic extrinsics are collected in the MoCap room to synchronize the robot arms' trajectories with the MoCap system. Discrete static trajectories and continuous static sequences are collected in indoor and outdoor environments. For outdoor environments, we record the static extrinsic in the MoCap system before moving to an outdoor environment. Because the robot arms are mounted on a wheeled platform, we can capture real-world factors, *e.g.,* vibrations, environmental dynamics, and weather in outdoor environments.

Table 2: CALIBBENCH Dataset Breakdown for Indoor (blue) and Outdoor Data (green).

Scene	Sequences	Samples	Type
MoCap	85	20,033	Indoor
Lab	10	2,005	Indoor
Patio	25	5,249	Outdoor
Outdoor Hallways	15	3,115	Outdoor
Parking Lot	20	4,354	Outdoor
Sidewalk/Road	15	3,686	Outdoor
Total	**170**	**38,442**	

Across the sequences collected in CALIBBENCH, Fig. 8 illustrates the diversity in extrinsics collected. We adopt real-world sensor configurations used in both autonomous driving and robotics. The LiDAR is offset from the camera within $[-2.05, 2.00]$, $[-1.82, 1.63]$, and $[-2.53, 1.45]$ meter in translation for $x-$, $y-$, and $z-$ axes. The dataset captures significant rotational variation, with rotations spanning $[-156°, 178°]$, $[-90°, 82°]$, and $[-180°, 180°]$ about the x-, y-, and z-axes, respectively. During discrete traces with dynamic extrinsics, the robot arm translates with an average velocity less than 0.04 m/s, minimizing LiDAR distortion.

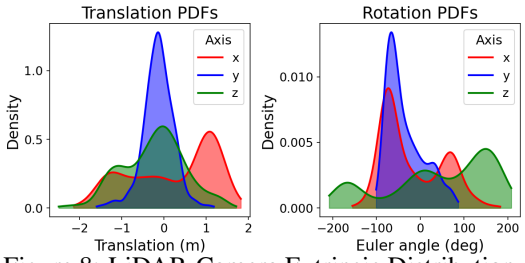


Figure 8: LiDAR-Camera Extrinsic Distribution.

5 EVALUATION AND BENCHMARK

Overview. We benchmark six recent cross-modal calibration methods on CALIBBENCH: Koide3 (Koide et al., 2023), DiffusionCalib (Ou et al., 2025), and ProjFusion (Ou et al., 2026), BEVCalib (Yuan et al., 2025), LCCRAFT Lee & Chen (2024), and CalibNet Iyer et al. (2018). All of these methods are targetless calibration methods. Koide3 is considered a hybrid approach since it leverages a vision model to extract features for zero-shot cross-modal calibration. The other methods are data-driven approaches. We use the official implementations of these methods. Appendix section E shows the poor generalization of these baselines' on CALIBBENCH when using their original KITTI or NuScenes checkpoints, and to demonstrate the utility of CALIBBENCH for training, we retrain the data-driven methods on CALIBBENCH from scratch and evaluate[1] them. During training, we use a training, validation, and test split of 0.8, 0.1, and 0.1 at the *sequence level*.

To benchmark the baseline methods on CALIBBENCH, we evaluate their performance on *discrete* traces with *static and dynamic* (*DS*, *DD*) extrinsics in indoor and outdoor settings. For continuous traces, the data-driven baselines process each frame independently, as designed. Koide3 instead requires a temporal history, assumes a fixed extrinsic throughout that history, and thus processes the

[1]For the initial guess, we perturb the ground-truth extrinsic with (± 0.5 m, $\pm 5°$) noise

Table 3: Indoor Benchmark for Discrete Static *(DS)* and Continuous Static *(CS)* Traces.

Trace	Method	Magnitude↓		Translation (cm) ↓			Rotation (°) ↓		
		E_t (cm)	E_R (°)	**X**	**Y**	**Z**	**Roll**	**Pitch**	**Yaw**
Discrete Static	ProjFusion (Ou et al., 2026)	41.3	1.7	22.7 ± 13.8	23.3 ± 14.7	20.7 ± 15.5	0.8 ± 1.1	0.8 ± 0.5	0.9 ± 1.0
	LCCRAFT (Lee & Chen, 2024)	34.7	3.1	20.4 ± 15.2	10.3 ± 9.3	20.3 ± 16.0	1.8 ± 0.9	1.5 ± 0.8	1.8 ± 1.3
	DiffusionCalib (Ou et al., 2025)	39.3	3.8	17.9 ± 9.0	19.0 ± 11.2	25.8 ± 14.8	1.6 ± 1.0	2.1 ± 1.0	2.4 ± 0.6
	CalibNet (Iyer et al., 2018)	28.1	2.7	13.4 ± 13.0	8.3 ± 8.3	17.4 ± 12.3	0.9 ± 0.6	1.6 ± 0.9	1.8 ± 1.0
	BEVCalib (Yuan et al., 2025)	24.1	4.5	8.6 ± 8.3	15.3 ± 11.5	11.9 ± 7.1	2.8 ± 2.0	2.4 ± 1.7	1.7 ± 1.2
Continuous Static	ProjFusion (Ou et al., 2026)	21.9	2.6	14.3 ± 10.7	8.5 ± 6.5	10.0 ± 7.4	1.4 ± 1.1	1.3 ± 1.1	1.3 ± 1.1
	LCCRAFT (Lee & Chen, 2024)	26.3	2.5	13.9 ± 10.6	10.6 ± 8.1	15.4 ± 9.7	1.3 ± 1.0	1.3 ± 1.1	1.3 ± 1.1
	DiffusionCalib (Ou et al., 2025)	17.4	2.4	11.9 ± 9.7	7.1 ± 6.4	6.4 ± 5.4	1.1 ± 0.9	1.4 ± 1.1	1.2 ± 1.0
	CalibNet (Iyer et al., 2018)	17.9	2.3	12.2 ± 10.7	6.9 ± 5.8	7.2 ± 5.9	1.0 ± 0.8	1.3 ± 1.1	1.2 ± 1.1
	BEVCalib (Yuan et al., 2025)	14.2	0.9	3.7 ± 3.5	12.4 ± 10.5	3.5 ± 3.4	0.4 ± 0.3	0.3 ± 0.4	0.6 ± 0.7
	Koide3 (Koide et al., 2023)	50.9	8.9	25.2 ± 17.9	22.8 ± 15.1	26.5 ± 22.6	4.4 ± 3.1	3.9 ± 3.5	4.9 ± 3.9

Table 4: Outdoor Benchmark for Discrete Static *(DS)* and Continuous Static *(CS)* Traces.

Trace	Method	Magnitude↓		Translation (cm) ↓			Rotation (°) ↓		
		E_t (cm)	E_R (°)	**X**	**Y**	**Z**	**Roll**	**Pitch**	**Yaw**
Discrete Static	ProjFusion (Ou et al., 2026)	36.2	2.2	9.6 ± 5.0	24.0 ± 23.9	18.0 ± 20.0	1.1 ± 0.6	1.1 ± 1.0	1.3 ± 0.6
	LCCRAFT (Lee & Chen, 2024)	43.2	2.1	29.9 ± 19.4	13.7 ± 12.3	18.2 ± 10.6	1.3 ± 0.8	1.1 ± 0.6	0.8 ± 0.6
	DiffusionCalib (Ou et al., 2025)	48.7	3.4	31.1 ± 29.8	19.3 ± 13.5	22.3 ± 18.6	2.2 ± 1.7	1.3 ± 1.3	1.0 ± 1.2
	CalibNet (Iyer et al., 2018)	28.8	2.1	14.4 ± 9.3	10.0 ± 10.1	16.3 ± 13.1	1.4 ± 0.6	1.2 ± 0.7	0.6 ± 0.6
	BEVCalib (Yuan et al., 2025)	31.4	3.5	13.1 ± 12.2	19.6 ± 15.0	17.0 ± 15.6	2.1 ± 0.9	2.2 ± 1.3	1.6 ± 1.0
Continuous Static	ProjFusion (Ou et al., 2026)	16.1	2.4	13.6 ± 9.9	2.9 ± 2.6	5.3 ± 4.0	1.3 ± 1.1	1.2 ± 1.1	1.2 ± 1.1
	LCCRAFT (Lee & Chen, 2024)	33.5	2.4	13.5 ± 10.0	11.5 ± 8.2	24.9 ± 13.9	1.2 ± 1.0	1.3 ± 1.1	1.2 ± 1.1
	DiffusionCalib (Ou et al., 2025)	19.7	2.5	13.2 ± 10.0	5.2 ± 4.1	10.5 ± 7.8	1.1 ± 0.8	1.3 ± 1.1	1.2 ± 1.0
	CalibNet (Iyer et al., 2018)	18.6	2.2	11.7 ± 10.3	5.9 ± 5.1	9.8 ± 7.9	1.0 ± 0.8	1.2 ± 1.1	1.2 ± 1.1
	BEVCalib (Yuan et al., 2025)	14.6	0.7	3.0 ± 3.2	13.2 ± 11.3	3.1 ± 3.2	0.4 ± 0.3	0.2 ± 0.1	0.5 ± 0.4
	Koide3 (Koide et al., 2023)	61.5	8.1	29.8 ± 16.0	27.7 ± 22.4	35.4 ± 29.7	4.6 ± 3.0	3.7 ± 2.9	3.8 ± 3.8

full continuous trace. To the best of our knowledge, there is no method that can take a dynamic extrinsic history to predict the extrinsics of the current frame; therefore, we only report Koide3's performance on *continuous* traces with *static* (*CS*) extrinsics. Sensors remain stationary in some continuous sequences to allow for point accumulation. We report the error magnitude of translation (E_t) and rotation (E_R) errors as the L_2 norm and geodesic distance between orientations, respectively, and we also report the axis-wise error (mean and standard deviation) as the absolute error between the predicted and ground-truth translation vectors and Euler angles.

Quantitative Benchmark Results. Due to the differences between indoor and outdoor scenes, we evaluate the baselines in these scenes separately in Table 3 and Table 4 on *DS* and *CS* traces. For reference (Appendix section E), we evaluated the baselines' original checkpoints trained on KITTI (Geiger et al., 2013) or NuScenes Caesar et al. (2020), and at the magnitude level, the original checkpoints transfer poorly to CalibBench, with errors reaching at most 181.5 cm and 24.1°. Training on CALIBBENCH substantially improves all learned baselines, but non-trivial error remains. *CS* traces are consistently easier than *DS* traces after training. Across indoor and outdoor scenes, all methods achieve lower E_t on continuous-static data, and rotation magnitudes are typically reduced to roughly 1–3°. This is consistent with continuous-static sequences providing repeated observations of the same extrinsic, whereas discrete-static takes provide only a single LiDAR–camera pair per extrinsic. Axis-wise results reflect the scalar magnitude results but provide more granular insight into failure modes. For example, BEVCalib has higher y-axis translation error in indoor and outdoor *DS* and *CS* traces (15.3, 12.4, 19.6, 13.2 cm), corresponding to the missing height dimension in the BEV representation. In contrast, CalibNet and LCCRAFT have the least error on the y-axis compared to other axes. ProjFusion has equivalent error of about 22–24 translation error across all axes in indoor *DS*, but its x-axis, or horizontal axis, error is higher in indoor and outdoor *CS* traces while having the least error in outdoor *DS* traces. DiffusionCalib also has the most error on the x-axis with the exception of indoor *DS* traces (17.9 cm). For rotation, most baselines have the same level

Table 5: Indoor Benchmark for Discrete Dynamic *(DD)* Traces.

Method	Magnitude↓		Translation (cm) ↓			Rotation (°) ↓		
	E_t (cm)	E_R (°)	**X**	**Y**	**Z**	**Roll**	**Pitch**	**Yaw**
ProjFusion (Ou et al., 2026)	33.9	2.5	15.7 ± 11.9	17.7 ± 14.9	18.2 ± 14.7	1.3 ± 1.1	1.3 ± 1.1	1.3 ± 1.1
LCCRAFT (Lee & Chen, 2024)	40.7	2.5	19.2 ± 13.7	16.3 ± 12.2	25.2 ± 16.7	1.3 ± 1.1	1.3 ± 1.1	1.3 ± 1.1
DiffusionCalib (Ou et al., 2025)	83.7	6.0	22.8 ± 21.2	51.4 ± 82.1	45.7 ± 22.5	4.7 ± 5.7	1.8 ± 1.6	2.0 ± 1.9
CalibNet (Iyer et al., 2018)	31.6	2.6	14.2 ± 11.6	15.9 ± 14.3	17.5 ± 12.3	1.4 ± 1.2	1.3 ± 1.1	1.3 ± 1.1
BEVCalib (Yuan et al., 2025)	22.2	3.2	8.5 ± 8.2	14.2 ± 11.4	10.0 ± 10.1	1.6 ± 1.2	1.5 ± 1.2	1.8 ± 1.3

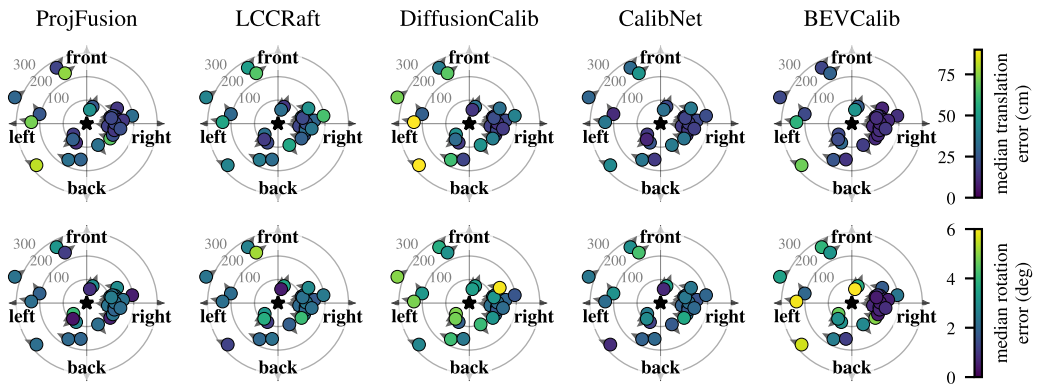


Figure 9: Correlation between sensor relative pose and translation (top) & rotation (bot) error. Center Star: LiDAR; Dots: camera. Arrow: orientation. Color: error magnitude

of error across all axes with a consistent standard deviation of about 1° in error. The hybrid baseline Koide3 remains substantially worse in both indoor and outdoor scenes, with translation errors on the order of the injected (± 0.5 m, $\pm 5^\circ$) perturbation, suggesting that it often fails to fully recover from the perturbation.

Discrete dynamic traces are collected only in the indoor MoCap scene and are evaluated separately in Table 5. Training the baselines on CALIBBENCH again provides a large improvement over their original checkpoints (Appendix section E), but the baselines have significant error on these traces with unique extrinsics, on-par with *DS* traces. BEVCalib improves slightly from 24.1 to 22.2 cm in E_t and from 4.5° to 3.2° in E_R, while ProjFusion reduces E_t from 41.3 to 33.9 cm but increases E_R from 1.7° to 2.5°. LCCRaft and CalibNet show modest increases in translation error, whereas DiffusionCalib degrades substantially from 39.3 to 83.7 cm and from 3.8° to 6.0°. Axis-wise errors follow the same trend, with most trained methods remaining near 1–2° per rotation axis, while DiffusionCalib exhibits notably larger variance. Overall, BEVCalib achieves the lowest trained translation error on the dynamic evaluation at 22.2 cm.

We further examine the correlation between sensor configuration, *i.e.,* where the camera is relative to the LiDAR, and extrinsic error. Fig. 9 shows that larger camera–LiDAR separation can increase translation error. ProjFusion has a few faraway examples to the left of the LiDAR in the 60-cm translation error range. DiffusionCalib also has 2 distant samples to the left of the LiDAR reaching 90 cm of error. Similarly, translation error grows as samples move farther away for LCCRAFT and CalibNet. In contrast, no consistent relationship is observed between relative orientation and rotation error. Both BEVCalib and DiffusionCalib have samples reaching 6° in rotation error, but these samples are near the LiDAR. Majority of samples tend to exhibit rotation error of 2° to 3°.

Result Validation The baselines perform much worse on CALIBBENCH compared to previous benchmarks. To verify the quality of our extrinsics and thus the CALIBBENCH results, we reproduce baselines on KITTI and reproduce baselines on a fixed-extrinsic subset of CALIBBENCH.

Table 6: **Dataset Comparison.** Trans. (cm) / Rot. ($^\circ$)

Method	KITTI Reported	KITTI Reproduced	CalibBench Fixed Extrinsic	CalibBench Diverse Extrinsics
ProjFusion	1.4 / 0.4	2.5 / 1.7	8.8 / 1.4	21.9 / 2.6
LCCRAFT	2.9 / 1.4	11.0 / 1.5	5.3 / 1.7	26.3 / 2.5
DiffusionCalib	5.4 / 2.8	5.6 / 4.1	3.0 / 1.3	17.4 / 2.4
CalibNet	4.9 / 0.4	2.8 / 1.6	7.7 / 2.4	17.9 / 2.3
BEVCalib	2.4 / 0.1	2.0 / 0.3	11.9 / 0.2	14.2 / 0.9
Koide3	–	51.4 / 4.5	49.0 / 6.1	50.9 / 8.9

Table 6 reports the baselines' results on KITTI from the literature and our reproduction of these results on KITTI. Additionally, we curate a CALIBBENCH subset, referred to as CALIBBENCH *Fixed Extrinsic*, with a similar number of frames as KITTI and has a single extrinsic as well, and we compare to the baselines' results on indoor *CS* traces, referred to as CALIBBENCH *Diverse Extrinsics*. The reproduction of all targetless baselines on KITTI achieves much smaller error, consistent with the reported results in their paper. Koide3, which was not reported on KITTI or uses our perturbation evaluation scheme, has a larger error that reflects the (± 0.5 m, $\pm 5^\circ$) perturbation across all dataset variants. In the CALIBBENCH *Fixed Extrinsic* subset, all data-driven baselines achieve errors comparable to those on KITTI and lower than on the CALIBBENCH *Diverse Extrinsics* subset. Given that we have reproduced the baselines' performance on KITTI and achieve similar performance in the same

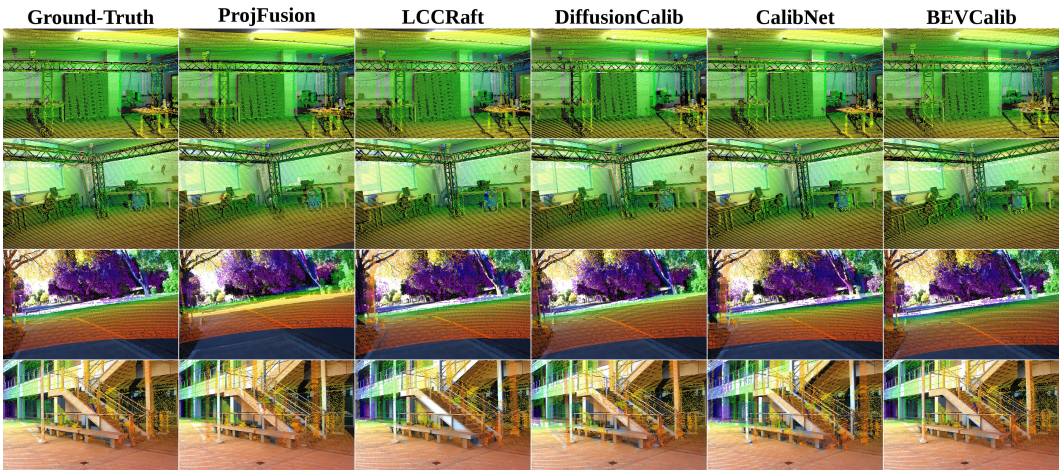


Figure 10: Qualitative Benchmark Results. The columns show, from left to right, the original image, the lidar-image overlay using CALIBBENCH's ground truth extrinsics, the lidar-image overlay using the extrinsics estimated by ProjFusion (Ou et al., 2026), LCCRaft (Lee & Chen, 2024), DiffusionCalib (Ou et al., 2025), CalibNet (Iyer et al., 2018) and BEVCalib (Yuan et al., 2025) trained on CALIBBENCH. First two rows are indoor traces. Last two rows are outdoor traces.

fixed-extrinsic setting, the baseline's results on CALIBBENCH in its default diverse extrinsic setting is indeed valid. Their weak performance on CALIBBENCH can be attributed to the inherent difficulty of having diverse extrinsics compared to prior benchmarks and reveals a research gap for future baselines to address.

Qualitative Benchmark Results. Fig. 10 illustrates examples of the corresponding qualitative results in the indoor setting. From left to right, it shows the original image from the dataset, the lidar-to-camera overlay using CALIBBENCH's ground-truth, and the overlays produced by the retrained baselines' predicted extrinsics. Overall, the overlays produced by the predicted extrinsics of the baselines are close to the ground-truth overlays with minor deviations. BEVCalib exhibits more deviation along the vertical dimension, consistent with the larger error on the y-axis given the lack of height in the BEV dimension. The other baselines achieve similar quality overlays. Across all scenes, ProjFusion and LCCRaft's predicted extrinsic results in a horizontal shift in the reprojected points, corresponding to a larger error in the x-axis. With the exception of the first indoor scene (top row), DiffusionCalib does not exhibit the same horizontal shift on these frames, but there is evidence of worse rotation prediction where points tilt relative to the corresponding object. LCCRaft has higher-quality reprojection overlays in the indoor scenes, but in outdoor scenes, there is a noticeable horizontal shift between objects and their corresponding LiDAR points.

6 CONCLUSION

In this work, we propose a diverse cross-modal calibration dataset CALIBBENCH. We collect a variety of traces that include *static and dynamic* extrinsics in *discrete and continuous* collections, in both indoor and outdoor scenarios. On CALIBBENCH, we train and benchmark several state-of-the-art methods. The results demonstrate that the current cross-modal calibration methods, both hybrid and data-driven, struggle to plug-and-play in arbitrary real-world systems. Training these methods on CALIBBENCH indeed improves the models' performance, but the remaining gap illustrates the gap that current methods and the inherent challenge of generalizing to diverse extrinsic that CALIBBENCH provides. In the future, we plan to reduce CALIBBENCH's reliance on MoCap systems by using robot arms' kinematics to calculate the sensor extrinsic, enabling easier data collection in more diverse scenarios. Beyond LiDAR and camera, we also hope to collect data from other sensors, *e.g.,* radar and thermal camera, and develop novel data-driven techniques using CALIBBENCH since robots and autonomous systems often have more than just these two sensor modalities. By providing a diverse and challenging CALIBBENCH, we hope to establish a training dataset and benchmark for future cross-modal calibration research, paving the way toward robust methods.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods, clean and reformat dataset, support qualitative and thematic data analysis, and interpret results. We have not used generative AI tools for proposing or refining hypotheses, assisting with translation, and designing or providing feedback on research methodology or experiments. Developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for creating/editing software code, creation of artifacts, draft parts of a research paper, and editing a research paper to improve readability. We have reviewed all AI-assisted work. Dataset quality and result analysis was verified by the authors. LLM-assisted paper writing was always used as a reference and edited for clarity and conciseness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11618–11628, 2020. doi: 10.1109/CVPR42600.2020.01164.

Filippo D'Addeo, Lorenzo Cipelli, Adriano Cardace, Emanuele Ghelfi, Andrea Zinelli, and Massimo Bertozzi. Calibbev: Lidar-camera calibration via bev alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4345–4354, March 2026.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

Quentin Herau, Nathan Piasco, Moussab Bennehar, Luis Roldao, Dzmitry Tsishkou, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. Soac: Spatio-temporal overlap-aware multi-sensor calibration using neural radiance fields. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15131–15140, 2024. doi: 10.1109/CVPR52733.2024.01433.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Jiunn-Kai Huang and Jessy W. Grizzle. Improvements to Target-Based 3D LiDAR to Camera Calibration. *IEEE Access*, 8:134101–134110, 2020. doi: 10.1109/ACCESS.2020.3010734.

Zhiwei Huang, Yikang Zhang, Qijun Chen, and Rui Fan. Online, target-free lidar-camera extrinsic calibration via cross-modal mask matching. *IEEE Transactions on Intelligent Vehicles*, pp. 1–12, 2024. doi: 10.1109/TIV.2024.3456299.

Ryoichi Ishikawa, Takeshi Oishi, and Katsushi Ikeuchi. Lidar and camera calibration using motion estimated by sensor fusion odometry, 2018. URL https://arxiv.org/abs/1804.05178.

Ganesh Iyer, R. Karnik Ram, J. Krishna Murthy, and K. Madhava Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2018. doi: 10.1109/iros.2018.8593693. URL http://dx.doi.org/10.1109/IROS.2018.8593693.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003, 2023. doi: 10.1109/ICCV51070.2023.00371.

Kenji Koide, Shuji Oishi, Masashi Yokozuka, and Atsuhiko Banno. General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11301–11307. IEEE, 2023.

Yu-Chen Lee and Kuan-Wen Chen. Lccraft: Lidar and camera calibration using recurrent all-pairs field transforms without precise initial guess. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16669–16675, 2024. doi: 10.1109/ICRA57147.2024.10610756.

Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V. Le, Alan Yuille, and Mingxing Tan. DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17161–17170, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01667. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01667.

Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2774–2781, 2023. doi: 10.1109/ICRA48891.2023.10160968.

Zhaotong Luo, Guohang Yan, Xinyu Cai, and Botian Shi. Zero-training lidar-camera extrinsic calibration method using segment anything model. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14472–14478, 2024. doi: 10.1109/ICRA57147.2024.10610983.

Xudong Lv, Boya Wang, Ziwen Dou, Dong Ye, and Shuo Wang. Lccnet: Lidar and camera self-calibration using cost volume network. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2888–2895, 2021. doi: 10.1109/CVPRW53098.2021.00324.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, December 2021. ISSN 0001-0782. doi: 10.1145/3503250. URL https://doi.org/10.1145/3503250.

Ni Ou, Zhuo Chen, Xinru Zhang, and Junzheng Wang. Iterative camera-lidar extrinsic optimization via surrogate diffusion. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3069–3075, 2025. doi: 10.1109/IROS60139.2025.11247343.

Ni Ou, Zhuo Chen, Xinru Zhang, and Junzheng Wang. Native-domain cross-attention for camera–lidar extrinsic calibration under large initial perturbations. *IEEE Robotics and Automation Letters*, 11(6):6983–6990, 2026. doi: 10.1109/LRA.2026.3683597.

Gaurav Pandey, James R. McBride, Silvio Savarese, and Ryan M. Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pp. 2053–2059. AAAI Press, 2012.

S. Remy, M. Dhome, J.M. Lavest, and N. Daucher. Hand-eye calibration. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, volume 2, pp. 1057–1065 vol.2, 1997. doi: 10.1109/IROS.1997.655141.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4937–4946, 2020. doi: 10.1109/CVPR42600.2020.00499.

Adarsh Jagan Sathyamoorthy, Jing Liang, Utsav Patel, Tianrui Guan, Rohan Chandra, and Dinesh Manocha. Densecavoid: Real-time navigation in dense crowds using anticipatory behaviors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11345–11352, 2020. doi: 10.1109/ICRA40945.2020.9197379.

Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1803–1810, 2017. doi: 10.1109/IVS.2017.7995968.

Ayon Sen, Gang Pan, Anton Mitrokhin, and Ashraful Islam. Scenecalib: Automatic targetless calibration of cameras and lidars in autonomous driving. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7771–7777, 2023. doi: 10.1109/ICRA48891.2023.10161316.

Jieying Shi, Ziheng Zhu, Jianhua Zhang, Ruyu Liu, Zhenhua Wang, Shengyong Chen, and Honghai Liu. Calibrcnn: Calibrating camera and lidar by recurrent convolutional neural network and geometric constraints. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10197–10202, 2020. doi: 10.1109/IROS45743.2020.9341147.

Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. doi: 10.1109/MRA.2012.2205651. https://ompl.kavrakilab.org.

Hanzhe Teng, Yipeng Wang, Xiaoao Song, and Konstantinos Karydis. Multimodal dataset for localization, mapping and crop monitoring in citrus tree farms. In *International Symposium on Visual Computing*, pp. 571–582. Springer Nature Switzerland Cham, 2023.

Shan Wu, Amnir Hadachi, Damien Vivet, and Yadu Prabhakar. Netcalib: A novel approach for lidar-camera auto-calibration based on deep learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 6648–6655, 2021. doi: 10.1109/ICPR48806.2021.9412653.

Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, Yunlong Wang, and Diange Yang. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 3095–3101, 2021. doi: 10.1109/ITSC48978.2021.9565009.

Guohang Yan, Feiyu He, Chunlei Shi, Pengjin Wei, Xinyu Cai, and Yikang Li. Joint camera intrinsic and lidar-camera extrinsic calibration. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11446–11452, 2023. doi: 10.1109/ICRA48891.2023.10160542.

Ze Yang, George Chen, Haowei Zhang, Kevin Ta, Ioan Andrei Bârsan, Daniel Murphy, Sivabalan Manivasagam, and Raquel Urtasun. Unical: Unified neural sensor calibration. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXVI*, pp. 327–345, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72763-4. doi: 10.1007/978-3-031-72764-1_19. URL https://doi.org/10.1007/978-3-031-72764-1_19.

Weiduo Yuan, Jerry Li, Justin Yue, Divyank Shah, Konstantinos Karydis, and Hang Qiu. Bevcalib: Lidar-camera calibration via geometry-guided bird's-eye view representations. In *9th Annual Conference on Robot Learning*, 2025.

Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

Qilong Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pp. 2301–2306 vol.3, 2004. doi: 10.1109/IROS.2004.1389752.

Morui Zhu. dCAP: Dynamic calibration and articulated perception for autonomous trucks. https://github.com/zhumorui/dCAP, 2026. GitHub repository; associated with the CVPR 2026 paper "Mind the Hitch: Dynamic Calibration and Articulated Perception for Autonomous Trucks".

A Dataset Release

We plan to release the dataset on HuggingFace under the CC BY 4.0 license. In the dataset release, we will include the raw data and processed data. The raw data will include the sensor streams, MoCap data, robot arm EE poses, and camera intrinsic. Fig. 11 shows the format of the processed CalibBench. Each sequence type is separate into take IDs and include the extrinsics in the `extrinsic.npz` binary. *Discrete static* sequences only have one extrinsic while the other sequences have N extrinsics depending on the sequence duration. Each extrinsic would have a corresponding image and point cloud pair that can be determined through the provided timestamp `<t>`.

```
calib_bench/
├── discrete-static/
├── continuous-static/
│   ├── take_1/
│   │   ├── extrinsic.npz
│   │   ├── data/image/
│   │   │   ├── image_<t>.png
│   │   │   └── ...
│   │   └── data/pcd/
│   │       ├── point_cloud_<t>.pcd
│   │       └── ...
│   ├── take_2/
│   │   └── ...
│   ├── take_3/
│   │   └── ...
│   ├── ⋮
│   └── take_N/
├── discrete-dynamic/
├── continuous-dynamic/
└── camera_intrinsics.npz
```

Figure 11: Dataset directory structure. The field `<t>` denotes the timestamp associated with each image and point cloud. Only the `continuous-static` category is expanded for clarity.

B Additional Data Collection Methodology

Diverse Sensor Extrinsics. Fig. 12 illustrates how markers are mounted on the robot arms' end effectors and the sensors themselves. As mentioned, markers are mounted asymmetrically to ensure precise tracking of the end effectors and sensors' rigid bodies. After calibrating the MoCap system, the MoCap system triangulating a rigid body can achieve an accuracy of *<0.11 millimeters*.

However, to reproduce realistic robot sensor placements in the real world, the arms' maneuver has to be carefully arranged with various geometric constraints in mind. On top of that, because the sensors are mounted behind the last joint, we need to keep the last joint fixed during the movement, such that the relative transform between the end-effector and the sensor is static. This is helpful because the sensor and end-effector can form a rigid body, enabling the robot arm to infer the sensor pose based on the internal effector pose (kinematics only).

To address this, we integrate the arms with ROS2 through the open-source ROS2 Kortex package and robustly control them via the MoveIt2 framework built upon the Open Motion Planning Library (OMPL Şucan et al. (2012)). Within this framework, we compute the inverse kinematics (IK) to determine the joint angles corresponding to specific end-effector poses. This allows for Cartesian control of the robot arm, enabling the end-effector to maintain a consistent orientation while stabilizing the attached sensor and moving it towards desired poses. For dynamic traces, we program and execute a set of rich, multi-directional arm motions designed to induce large variability in sensor–robot extrinsics.

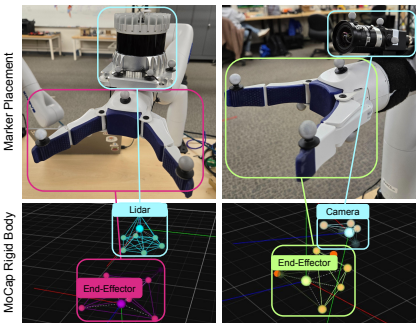


Figure 12: Marker placement on LiDAR, camera, and robot arm end-effectors, and their corresponding rigid body defined in the MoCap system.

Data Synchronization. The MoCap system records time-stamped poses of the arm's base and end-effector, producing a continuous trajectory $x(t)$, while the ROS system simultaneously logs its own kinematic transforms, producing trajectory $y(t)$. Because these streams run at different frequencies and with different clock offsets, precise temporal synchronization is required to obtain high-quality ground-truth correspondences.

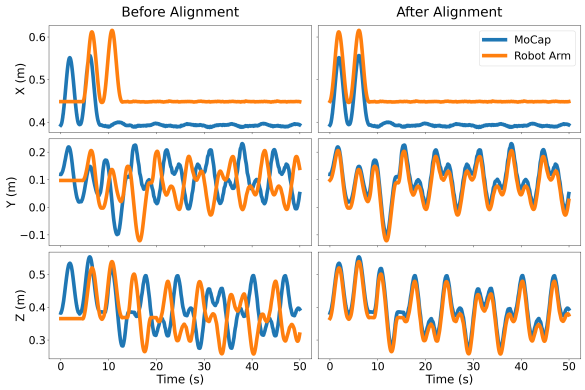


Figure 13: MoCap and ROS Time Sync

To estimate the relative clock offset, we compute time shift τ that best aligns the dynamic motion patterns of both trajectories. We apply a cross-correlation procedure in the frequency domain using Fast Fourier Transform (FFT), and solve for the delay τ such that $x(t-\tau) \approx y(t)$.

Fig. 13 shows the resulting trajectories before and after temporal alignment. Prior to alignment, the MoCap (blue) and robot arm (orange) signals exhibit visible phase discrepancies across all three axes. After applying the estimated τ, the trajectories become tightly matched, demonstrating that the cross-correlation-based synchronization effectively resolves clock drift.

C LiDAR Eye-hand Calibration

For eye-hand calibration we use ChArUco boards and AprilTags as the fiducial targets for the camera and LiDAR, respectively. The camera recovers its target pose from PnP on the detected ChArUco corners; the LiDAR, lacking photometric detail, instead yields noisy corners from its reflectivity image, motivating the geometry-based extraction we describe next. We therefore recover the four corners purely from point-cloud geometry. The border around the AprilTag is white, and its high reflectivity relative to the darker tag interior makes it stand out in the point cloud (a representative scene is shown in Fig. 14), yielding four crisp corners that serve as the LiDAR-side target features. These corners form the LiDAR's $B = \mathcal{T}_{\text{target}}^{\text{sensor}}$; the remainder of this section describes how we extract them and solve for X, summarised in Fig. 15.

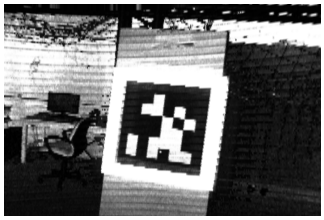


Figure 14: Example LiDAR reflectivity image of the scene. The white border around the AprilTag saturates bright, standing out from the rest of the scene.

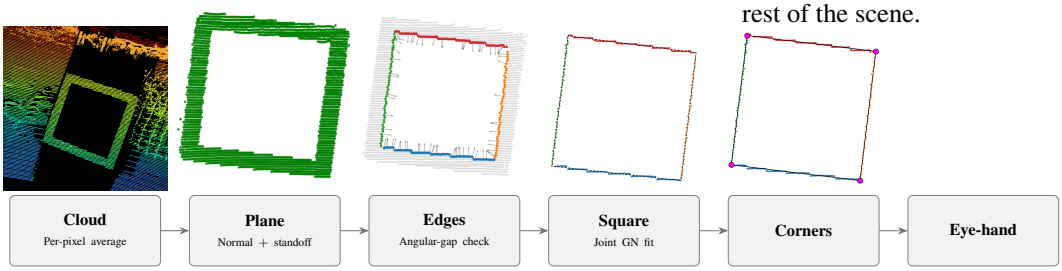


Figure 15: LiDAR corner-extraction pipeline. The averaged cloud is plane-fit to fix the board's tilt and standoff (3 DoF); the square's four edges are detected, fit jointly, and its corners recover the remaining in-plane rotation and position (3 DoF). These corner features determine B and are solved against all takes for the eye-hand transform X.

Cloud. Following the first stage of Fig. 15, for each take is static, so we collapse the capture into a single clean cloud by averaging x, y, z, and reflectivity per sensor pixel across all frames, suppressing per-frame noise before extraction of corners.

Plane. From this averaged cloud, we threshold by reflectivity, keeping the brightest returns within $5\,\text{m}$ to isolate the white border. The largest DBSCAN cluster ($\epsilon = 3\,\text{cm}$) is retained, and a RANSAC plane fit yields the normal, offset, and inliers, which we then grow along the plane to recover the full surface. This fixes three degrees of freedom, the orientation and its perpendicular standoff, but leaves the rotation about the normal, and the position within the plane, undetermined.

Inner Edges. To recover the remaining degrees of freedom, we turn to the square formed by the white area around the AprilTag; its four sides yield the corners that fix the in-plane rotation and

position. We therefore begin by extracting the sides. Working in the plane frame from the previous step, we project the plane inliers into the plane's (u, v) coordinates and identify edge points by a neighbour angular-gap check; for each point $\mathbf{q}$ we sort the bearings to its 25 nearest in-plane neighbours. An interior point is surrounded on all sides, whereas an edge point has one large gap facing the empty square interior, where the largest gap $g(\mathbf{q}) = \max_j(\beta_{(j+1)} - \beta_{(j)})$ exceeds $100°$. The bisector of that gap defines the point's *empty direction* $\hat{\mathbf{d}}(\mathbf{q})$ (arrows in Fig. 15); a point is accepted as an edge point if $\hat{\mathbf{d}}$ points toward the square's interior, self-assigning to a side (L, R, T, B, coloured in Fig. 15) from the sign and dominant axis of $\hat{\mathbf{d}}$. Each side is then fitted by robust least squares, iteratively rejecting points beyond $2.5\times$ the median absolute deviation (MAD) of the current line's residuals, yielding four clean edge point sets, one per side.

Square. Intersecting the four edge lines independently allows per-edge noise to skew the result, so we instead fit all four edges jointly. Each side $s \in \{L, R, T, B\}$ is a line $\mathbf{n}_s \cdot \mathbf{x} = \rho_s$ with unit normal $\mathbf{n}_s = (\cos\varphi_s, \sin\varphi_s)$, contributing an orientation angle φ_s and signed offset ρ_s; stacking the four sides gives the eight unknowns $\boldsymbol{\theta} = [\varphi_L, \rho_L, \varphi_R, \rho_R, \varphi_T, \rho_T, \varphi_B, \rho_B]^\top$.

Seeded from the per-side fits, $\boldsymbol{\theta}$ is refined with Gauss-Newton, jointly fitting the lines to their points while enforcing the square's known geometry. The objective stacks a Huber-weighted data term with three shape constraints parallel, perpendicular, and equal-side:

$$\begin{aligned}
r^{\text{data}}_{s,k} &= \sqrt{w_H(e_{s,k})}\, e_{s,k}, \quad e_{s,k} = \mathbf{n}_s \cdot \mathbf{p}_k - \rho_s, \;\; \mathbf{p}_k \in \mathcal{I}_s && \text{(point-to-line fit)} \\
r^{\parallel} &= \sqrt{M}\, \ell \left[\sin(\varphi_L - \varphi_R),\; \sin(\varphi_T - \varphi_B) \right]^\top && \text{(opposite sides parallel)} \\
r^{\perp} &= \sqrt{M}\, \ell \left[\cos(\varphi_L - \varphi_T),\; \cos(\varphi_R - \varphi_B) \right]^\top && \text{(adjacent sides perpendicular)} \\
r^{\text{eq}} &= \sqrt{M}\, (w_{LR} - w_{TB}) && \text{(equal side lengths)}
\end{aligned} \tag{1}$$

where $\mathcal{I}_s$ are the inliers of side s, M the total inlier count, ℓ the half-side length, and w_{LR}, w_{TB} the perpendicular distances between the opposite side pairs (L, R) and (T, B). The two perpendicular terms enforce right angles at adjacent corners, keeping the residual symmetric across the four corners. The data term uses the Huber weight $w_H(e) = 1$ for $|e| \le \delta$ and $\delta/|e|$ otherwise (with $\delta = 8\,\text{mm}$), so stray points cannot dominate. The parallel and perpendicular terms constrain the shape to a rectangle; the equal-side term makes it a square. To keep the few constraint rows from being outweighed by the hundreds of data rows, each is weighted by $\sqrt{M}$, and the angular terms scaled by ℓ so that radian and metre valued residuals are commensurate. The optimisation converges in a few iterations.

Corners. The joint fit yields a single square whose four corners $\{\mathbf{x}_m\}_{m=1}^4$ lie in the plane's (u, v) coordinates. Lifting them back into the LiDAR frame gives the corner observations $\mathbf{c}^m_{l,i}$ for take i the LiDAR-side target corners that constitute B.

Finally, we solve the LiDAR eye-hand transform $X \equiv \mathcal{T}_L$ by nonlinear least squares. For every take i the MoCap supplies the sensor rigid-body pose $\mathcal{T}^{\text{mocap}}_{\text{markers}_i}$, and the lifted corners $\mathbf{c}^m_{l,i}$ are the observations in the LiDAR frame. We minimise $\sum_i \sum_m \| \mathcal{T}^{\text{mocap}}_{\text{markers}_i} X\, \mathbf{c}^m_{l,i} - \mathbf{p}^m_w \|^2$ over X and the shared corner points $\{\mathbf{p}^m_w\}$, with X optimised on $\text{SE}(3)$. The recovered transform then feeds the reprojection validation, where the per-take estimates are averaged into a single target pose and the predicted corners reproject with a mean error of 2.77 ± 1.01 px (median 2.58, range 1.10–4.82).

Table 7: Evaluation of the ***original*** baseline checkpoints.

Scene	Trace	Method	Magnitude↓		Translation (cm) ↓			Rotation (°) ↓		
			E_t (cm)	E_R (°)	**X**	**Y**	**Z**	**Roll**	**Pitch**	**Yaw**
Indoor	Discrete Static	ProjFusion (Ou et al., 2026)	63.5	4.5	7.9 ± 6.1	50.7 ± 41.1	33.8 ± 16.8	2.2 ± 0.7	2.7 ± 1.8	2.3 ± 1.5
		LCCRAFT (Lee & Chen, 2024)	26.0	4.3	14.1 ± 7.6	12.4 ± 8.3	13.9 ± 6.4	3.0 ± 1.8	1.6 ± 1.2	1.7 ± 1.8
		DiffusionCalib (Ou et al., 2025)	88.3	20.9	27.9 ± 29.2	44.5 ± 33.6	68.1 ± 51.0	9.0 ± 1.0	4.0 ± 2.8	17.4 ± 9.6
		CalibNet (Iyer et al., 2018)	68.7	4.2	42.9 ± 22.4	26.8 ± 20.8	35.6 ± 22.5	2.0 ± 1.7	2.4 ± 2.0	2.0 ± 1.5
		BEVCalib (Yuan et al., 2025)	127.8	3.9	125.5 ± 47.6	6.9 ± 6.6	14.3 ± 12.6	1.2 ± 1.0	2.6 ± 1.7	1.8 ± 1.0
	Continuous Static	ProjFusion (Ou et al., 2026)	66.8	3.7	30.6 ± 23.2	37.3 ± 32.2	35.7 ± 23.0	2.0 ± 1.5	1.9 ± 1.6	1.6 ± 1.3
		LCCRAFT (Lee & Chen, 2024)	54.0	4.2	21.0 ± 14.5	32.2 ± 17.5	29.2 ± 20.5	3.0 ± 1.8	1.6 ± 1.3	1.6 ± 1.2
		DiffusionCalib (Ou et al., 2025)	149.4	11.2	29.3 ± 23.9	78.8 ± 33.0	119.6 ± 45.6	8.8 ± 4.8	2.7 ± 2.3	4.4 ± 3.6
		CalibNet (Iyer et al., 2018)	61.3	4.2	32.5 ± 22.7	30.0 ± 17.6	34.5 ± 15.4	2.8 ± 1.8	1.7 ± 1.3	1.8 ± 1.4
		BEVCalib (Yuan et al., 2025)	150.6	2.9	147.5 ± 60.4	13.5 ± 11.6	15.4 ± 11.6	1.4 ± 1.2	1.7 ± 1.3	1.3 ± 1.1
	Discrete Dynamic	ProjFusion (Ou et al., 2026)	64.5	3.5	31.4 ± 22.4	30.8 ± 20.3	37.4 ± 25.6	2.1 ± 1.8	1.5 ± 1.3	1.8 ± 1.5
		LCCRAFT (Lee & Chen, 2024)	45.0	4.5	20.6 ± 14.6	22.4 ± 16.0	25.5 ± 15.4	2.2 ± 1.5	1.4 ± 1.1	2.9 ± 2.0
		DiffusionCalib (Ou et al., 2025)	167.0	24.1	39.5 ± 28.8	54.0 ± 43.0	138.6 ± 71.7	11.6 ± 6.3	3.2 ± 2.7	18.5 ± 14.3
		CalibNet (Iyer et al., 2018)	52.6	6.3	26.3 ± 18.6	26.6 ± 17.8	28.1 ± 17.5	3.1 ± 2.1	2.5 ± 2.0	3.8 ± 2.9
		BEVCalib (Yuan et al., 2025)	131.2	3.3	127.1 ± 66.8	14.2 ± 11.4	14.8 ± 11.5	1.5 ± 1.2	2.1 ± 1.5	1.4 ± 1.2
Outdoor	Discrete Static	ProjFusion (Ou et al., 2026)	57.9	3.5	35.9 ± 22.6	25.8 ± 20.8	26.2 ± 14.1	1.6 ± 1.5	1.8 ± 1.2	2.1 ± 0.6
		LCCRAFT (Lee & Chen, 2024)	44.2	4.2	21.2 ± 15.4	22.7 ± 12.9	25.5 ± 15.5	2.2 ± 1.7	1.1 ± 1.1	2.6 ± 2.1
		DiffusionCalib (Ou et al., 2025)	108.0	11.9	42.2 ± 25.0	51.2 ± 36.6	78.5 ± 33.8	5.7 ± 4.4	4.4 ± 3.4	8.5 ± 5.7
		CalibNet (Iyer et al., 2018)	41.4	3.7	25.6 ± 16.3	18.8 ± 15.3	21.7 ± 14.0	2.2 ± 1.0	1.1 ± 1.1	2.4 ± 1.3
		BEVCalib (Yuan et al., 2025)	146.1	3.3	143.8 ± 25.1	14.3 ± 8.7	13.3 ± 14.2	1.7 ± 1.4	2.3 ± 1.2	1.2 ± 0.4
	Continuous Static	ProjFusion (Ou et al., 2026)	70.3	5.0	49.3 ± 34.4	23.3 ± 24.4	30.6 ± 18.4	2.3 ± 1.9	1.9 ± 1.4	3.2 ± 2.5
		LCCRAFT (Lee & Chen, 2024)	56.9	4.8	21.6 ± 15.1	35.5 ± 18.0	31.6 ± 17.5	3.8 ± 1.9	1.6 ± 1.2	1.6 ± 1.3
		DiffusionCalib (Ou et al., 2025)	181.5	18.2	62.1 ± 40.0	92.5 ± 53.1	123.7 ± 58.1	13.2 ± 5.6	6.1 ± 3.9	7.3 ± 8.4
		CalibNet (Iyer et al., 2018)	58.1	5.6	35.5 ± 20.0	33.4 ± 20.3	20.9 ± 15.0	4.5 ± 1.9	1.7 ± 1.4	2.0 ± 1.6
		BEVCalib (Yuan et al., 2025)	163.7	2.7	161.7 ± 20.9	13.0 ± 10.8	15.6 ± 11.4	1.3 ± 1.1	1.6 ± 1.2	1.3 ± 1.1

D Qualitative Reprojection Analysis

Fig. 16 shows examples comparing the reprojections before and after eye-hand calibration for the camera using a Charuco board and the LiDAR using an AprilTag. Because the LiDAR has no image plane, we project the predicted 3D corners (in the LiDAR frame) into a virtual pinhole rendering of the reflectivity image, so its reprojection error is measured in pixels just like the camera's. In the left column without eye-hand calibration, the projected fiducial targets' corners drift away from the corners. For Camera (top), this is shown by the distance (outlined in yellow) between the detected corners (red) and projected corners (green). For LiDAR (bottom), this is shown by the misaligned bounding box (red). After applying the eye-hand transforms, the reprojections in the right column align with their respective corners. Fig. 3 visualizes a LiDAR point cloud overlaid on top of a camera image after applying the eye-hand transforms on both sensors. The point cloud corresponding to the metal truss structure, desks, chairs, and printers overlay well on the objects in the scene. By achieving low reprojection error and observing the LiDAR-camera overlay, the calculated extrinsics are validated as ground-truth extrinsics.

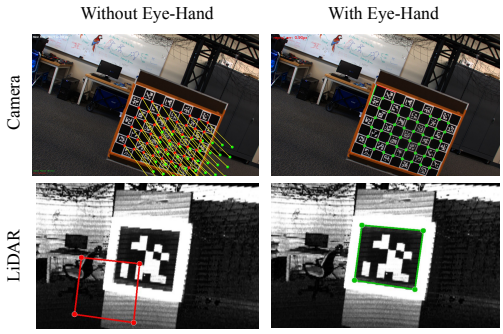
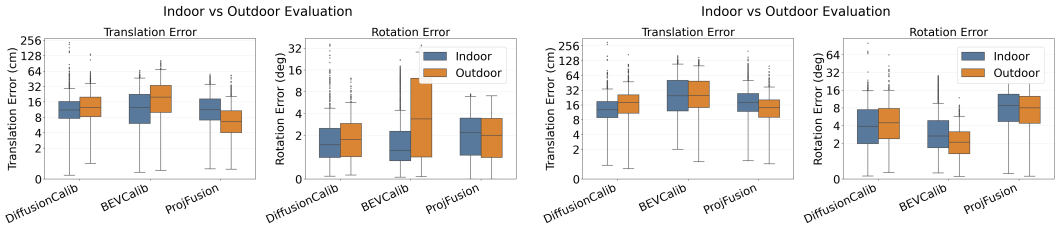


Figure 16: Reprojection Qualitative Analysis. Left Column: Before eye-hand calibration, Right Column: After eye-hand calibration.

E Additional Quantitative Evaluation

Table 7 shows the results from evaluating the data-driven baselines' original checkpoints, the ones trained on NuScenes (Caesar et al., 2020) or KITTI (Geiger et al., 2013), on CalibBench. Overall, the baselines generalize poorly to CalibBench's diverse extrinsics. In terms of rotation magnitude, the majority of the baselines average $4.1°$ of error with the exception of DiffusionCalib, which has an average rotation error of $17.26°$. In contrast, the baselines struggle more in predicting the extrinsics' correct translation component as shown by how their translation error magnitude often exceed 50 cm in error. Across the extrinsic traces and in both indoor and outdoor scenarios, BEVCalib and DiffusionCalib exceeds 100 cm in error. The axis-wise breakdown reflects their large magnitude error counterparts. From these results, we may conclude that CalibBench is indeed more difficult compared to prior benchmarks, and current cross-modal calibration methods are not equipped to predict diverse extrinsics accurately enough.



(a) $5°, 0.5$m perturbation (b) $20°, 1.5$m perturbation

Figure 17: Indoor vs Outdoor Performance Comparison.

F Environmental Bias

In this section, we directly compare the performance of the trained baselines on CalibBench between indoor and outdoor scenes to test for environmental bias. In indoor scenes such as the MoCap room, there are objects scattered around the room that makes the indoor data more complex and potentially more challenging in predicting accurate extrinsics. In contrast, we collect data in cleaner environments with unique and uniform patterns (*e.g.,* wall with uniform pattern, wall with windows) in CalibBench's outdoor scenes such as the patio and outdoor hallways, allowing us to compare if more complex visual features have an effect on the baselFig. 17 shows the distribution error on log-base 2 scale between the indoor and outdoor scenes with simpler features. In general, most baselines have a similar error distributions for both translation and rotation regardless of indoor and outdoor data. One exception is BEVCalib having a slighly wider range of rotation error in the outdoor setting compared to the indoor setting for $(5°, 0.5\text{m})$ perturbation. In addition, many outliers are present, suggesting that the baselines still struggle with certain samples. Overall, all baselines find indoor and outdoor scenes equally difficult, and environmental factors do not significantly impact calibration quality.

Multi-Camera Extrinsics In addition to multimodal extrinsics, CalibBench's setup allows for finding extrinsics between cameras. To demonstrate the feasibility, Fig. 18 calculates the extrinsic $\mathcal{T}_{C_1}^{C_2}$ between cameras C_1 and C_2. $\mathcal{T}_{C_1}^{C_2}$ is then used to transform and project the points detected in the image of C_1 into the image of C_2. We present the qualitative reprojection analysis using an April Tag to show the utility of repurposing CalibBench for training and benchmarking multi-camera calibration methods.

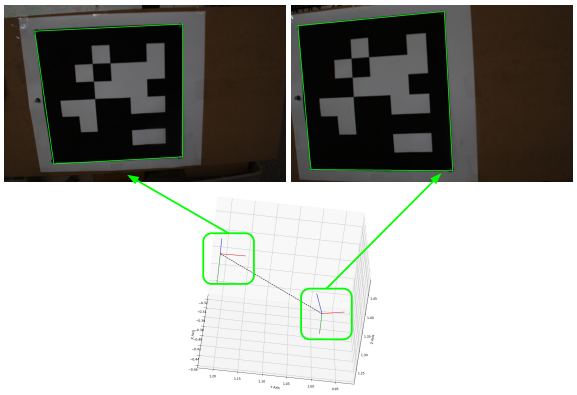


Figure 18: Successful Reprojection across Cameras