BEVCALIB: LiDAR-Camera Calibration via Geometry-Guided Bird's-Eye View Representations

Weiduo Yuan[*1], **Jerry Li**[*2], **Justin Yue**[2], **Divyank Shah**[2], **Konstantinos Karydis**[2], **Hang Qiu**[2]
[1] University of Southern California, [2] University of California, Riverside

Abstract: Accurate LiDAR-camera calibration is fundamental to fusing multi-modal perception in autonomous driving and robotic systems. Traditional calibration methods require extensive data collection in controlled environments and cannot compensate for the transformation changes during the vehicle/robot movement. In this paper, we propose the first model that uses bird's-eye view (BEV) features to perform LiDAR camera calibration from raw data, termed BEVCALIB. To achieve this, we extract camera BEV features and LiDAR BEV features separately and fuse them into a shared BEV feature space. To fully utilize the geometric information from the BEV feature, we introduce a novel feature selector to filter the most important features in the transformation decoder, which reduces memory consumption and enables efficient training. Extensive evaluations on KITTI, NuScenes, and our own dataset demonstrate that BEVCALIB establishes a new state of the art. Under various noise conditions, BEVCALIB outperforms the *best baseline in the literature by an average of (47.08%, 82.32%) on KITTI dataset, and (78.17%, 68.29%) on NuScenes dataset*, in terms of (translation, rotation), respectively. In the open-source domain, it improves the *best reproducible baseline by one order of magnitude*. Our code and demo results are available at https://cisl.ucr.edu/BEVCalib.

Keywords: LiDAR-Camera Calibration, Autonomous Driving, BEV Features

1 Introduction

Multi-modal sensing has been widely deployed in today's autonomous systems to provide accurate perception while adding redundancy for safety-critical applications. Previous work has shown improved reliability and effectiveness of multi-modal perception for navigation in crowded environments [1] and autonomous driving [2, 3] as a result of different sensing modalities complementing each other. One key enabler is the multimodal calibration that ensures the geometric alignment among different modalities. An extrinsic calibration error of a few degrees in rotation or a few cm in translation can compound over a distance (*e.g.,* a 20 cm displacement over 5 meters [4]), which can significantly degrade the performance of downstream tasks.

Early works in multimodal calibration relied on targets with unique planar patterns [5, 6, 4] or specialized rooms [7] as a reference to ensure proper geometry when aligning multiple modalities, primarily image and LiDAR modalities. Although effective, the usage of specialized equipment can make the calibration process tedious and cumbersome. Nevertheless, there is also a demand in modern autonomous systems for continuous calibration in the wild (*e.g.,* misoriented/shaken sensors). Consequently, other works focus on targetless approaches [8, 9], *e.g.,* relying on the motion of the sensors, using natural features in the environment. The advent of deep learning has further diversified the approaches taken for multimodal calibration. Some calibration methods are hybrid [10, 11], *i.e.,* they use deep learning models to extract features in different modalities and perform traditional optimization to predict the sensor extrinsics. Other methods [8, 12, 13] are purely data-driven and are trained and evaluated with popular datasets. such as KITTI [7], NuScenes [14] and Pandaset [15].

[*]Equal contribution. Email: weiduoyu@usc.edu, {jli793, jyue009, dshah048, karydis, hangq}@ucr.edu

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

Among these learning-based methods, a common pattern is to rely on techniques akin to feature matching between the images and the point clouds. Previous attempts to find these correspondences use feature matching models [10], segmentation masks [11], or the latent space after encoding images and point clouds as depth images [8, 9, 16, 17]. While useful for calibration, establishing correspondences does not explicitly enforce geometric constraints. In multi-modal perception works, one appealing method is the bird's-eye-view (BEV) representations [2] that place different modalities in a shared BEV grid. In this BEV grid, LiDAR point clouds are projected or pillarized onto the BEV grid, while camera features are also lifted into this space. Intrinsically, BEV representations preserve the geometry information, which offers a much stronger space for feature alignment. Such alignment has seen great success in various autonomous driving tasks, including object detection [18, 19, 20], HD-map construction [21, 22], place recognition [23, 24], occupancy [25, 26], and world model [27, 28]. We investigate BEV efficacy for geometric alignment and calibration.

In this work, we propose BEVCALIB, the first-of-its-kind target-less LiDAR-camera calibration method using BEV representations. This method is motivated by the need to explicitly ensure that geometry is maintained during the calibration process. To that end, BEVCALIB projects both an input image and a point cloud using an initial guess extrinsic T_{init} into BEV feature space, fuses these BEV features together, and follows a geometry-guided approach to decode T_{pred}, the correction needed to arrive at an accurate extrinsic transform. While we train BEVCALIB on KITTI and NuScenes for fair comparison with existing baselines, we also collect our own dataset (CALIBDB) with heterogeneous extrinsics to evaluate the generalizability. Our evaluation shows that BEVCALIB establishes a new state-of-the-art performance. Under various noise conditions, BEVCALIB outperforms the *best baseline in literature* by an average of (47.08%, 82.32%) on KITTI dataset, and (78.17%, 68.29%) on NuScenes dataset in terms of (translation, rotation) respectively. Compared to open source baselines, BEVCALIB outperforms the best reproduced results by (92.75%, 89.22%) on KITTI dataset, (92.69%, 93.62%) on NuScenes dataset, and (60.21%, 24.99%) on CALIBDB. Qualitative visualizations in the form of camera-LiDAR overlays illustrate a fine-grained projection match as a result of the higher accuracy of BEVCALIB's predicted extrinsics. With strong performance, BEVCALIB fills a critical gap in the open-source community for LiDAR-camera calibration. Our code and demo results are available at `https://cisl.ucr.edu/BEVCalib`.

2 Related Works

Target-based Methods. Early multimodal calibration methods borrowed from camera calibration techniques using planar targets, *e.g.,* checkerboards, fiducial markers, and other specialized patterns, to provide a reference in aligning modalities. Earlier works [5] found that LiDAR scans on the planar pattern can be used to register constraints with the estimated pattern on the camera's image plane, thus improving the extrinsic calibration of previous methods. Huang *et al.* [4] similarly found that using a target of known geometry and dimensions is helpful and developed a solution to fit the LiDAR to camera transform without requiring target edge extraction. Yan *et al.* [6] provides a way to jointly calibrate camera intrinsics and LiDAR to camera extrinsics using a special target type with checkerboards and conic sections. Verma *et al.* [29] proposed using a Variability of Quality (VOQ) metric to filter calibration samples to reduce user error and possible overfitting to the target.

Target-less Methods. While specialized targets ensure accuracy in predicting the sensor extrinsic, performing the sensor setup can be cumbersome and tedious. These drawbacks can be alleviated using target-less calibration methods. For example, Ishikawa *et al.* [30] proposed using motion, while Pandey *et al.* [31] proposed incorporating probabilistic methods in extrinsic calibration. Recent years have witnessed the rise of interest in solving calibration using learning-based approaches that leverage natural cues in the target-less setting. Interestingly, the literature follows a divergence of two approaches: combining neural networks with classical methods (*i.e.,* hybrid approach), and pure data-driven methods. Hybrid methods [10, 11] use neural networks (*e.g.,* SuperGlue [32, 33]) to perform feature extraction before predicting the sensor extrinsic through classical optimization methods. Another recent hybrid approach, MDPCalib [34], utilizes sensor motion estimates as coarse registration, followed by neural network prediction of 2D-3D correspondence for calibration refinement. On the other hand, data-driven methods train and evaluate neural networks on datasets

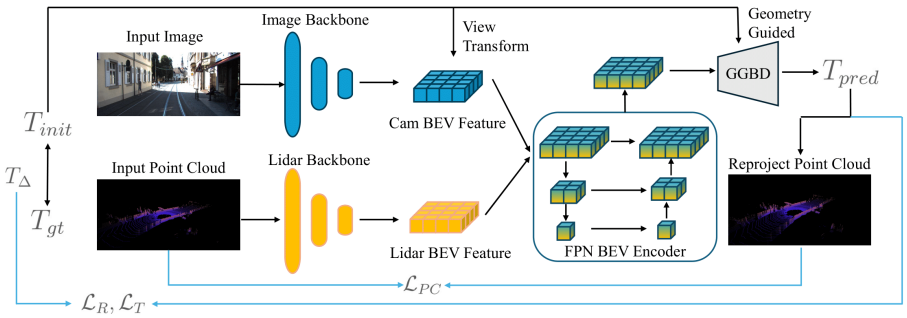


Figure 1: **Overall architecture of BEVCALIB.** The overall pipeline of our model consists of BEV feature extraction, FPN BEV Encoder, and geometry-guided BEV decoder (GGBD). For BEV feature extraction (§3.2), the inputs of the camera and LiDAR are extracted into BEV features through different backbones separately, then fused into a shared BEV feature space. The FPN BEV encoder is used to improve the multi-scale geometric information of the BEV representations. For geometry-guided BEV decoder (§3.3) utilizes a novel feature selector that efficiently decodes calibration parameters from BEV features. $\mathcal{L}_R$, $\mathcal{L}_T$, and $\mathcal{L}_{PC}$ are loss functions introduced at §3.4.

such as KITTI [7] and NuScenes [14]. Early learning-based methods [8, 9] encoded the image and LiDAR point cloud before treating the extrinsic prediction as a regression problem. More recently, some works [13, 35] use neural radiance fields (NeRF [36, 37]) as pseudo-targets to ensure explicit geometric alignment between the image and point cloud representations. 3D Gaussian Splatting [38] has also been exploited [39] for calibration with more efficient training compared to NeRF.

Bird's-eye View Feature. Bird's-eye view feature space has been used [40, 41] to structure 3D sensor data of the environment into a 2D feature plane. It provides a framework to efficiently extract features from an individual modality [18, 42, 43] or multiple modalities [2] based on geometric alignment. In recent works, the BEV feature has been adopted to address a wide range of tasks, such as object detection [19, 20], HD-map construction [21, 22], place recognition [23, 24], occupancy perception [25, 26], and world model [27, 28]. These works demonstrate the great potential of BEV features on various tasks. Compared to previous works [9, 17] that use a mis-calibrated depth image as LiDAR input, the BEV feature offers a more accurate and structured geometric representation. The closest to our work is CalibRBEV [44], but it only focuses on cameras. The work encodes detection bounding boxes into a BEV representation and applies cross-attention with image features to predict calibration parameters. However, using bounding boxes offers a strong prior knowledge that over-simplifies the calibration process. In contrast, BEVCALIB calibrates from raw LiDAR data, preserving more information for accuracy and robustness to corner cases. To the best of our knowledge, BEVCALIB is the first LiDAR-camera extrinsic calibration model using BEV features.

3 Methodology

3.1 Architecture Overview

BEVCALIB is designed as a target-less LiDAR-camera calibration model that takes a scene consisting of a single image and the full-scene LiDAR data as input and predicts the calibration parameters from LiDAR to camera. Figure 1 shows an overall architecture of BEVCALIB. It first extracts modality-specific 3D features from camera images and LiDAR using separate backbones (§3.2). These features are then projected and fused into a unified BEV representation to capture both semantic and geometric information. To enhance the BEV's spatial capability, we aggregate multi-scale features by a Feature Pyramid Network (FPN) BEV Encoder. Next, we propose a novel Geometry-Guided BEV feature Decoder (GGBD, §3.3). It first employs a geometry-guided feature selector guided by the coordinates derived from 3D image features, allowing the model to focus on spatially meaningful regions. Finally, it incorporates a refinement module to decode calibration parameters from selected features for efficient and effective training. Following the convention of learning-based calibration methods [45, 9], Table 1 summarizes the notations to describe our method.

Specifically, the image branch of BEVCALIB takes image input I, and utilizes T_{init} and K to generate a 3D frustum feature F_C^{3D} (see more details in §3.2). Simultaneously, the LiDAR branch

Table 1: Notation Summary

Symbol	Dimension	Description
I	$\mathbb{R}^{H\times W\times 3}$	RGB image captured by camera
P	$\mathbb{R}^{N\times 3}$	Point clouds captured by lidar, where $P_i = [X_i, Y_i, Z_i]$
K	$\mathbb{R}^{4\times 4}$	Intrinsic matrix of camera
T_{gt}	$\mathbb{R}^{4\times 4}$	Ground truth transformation from lidar to camera
T_Δ	$\mathbb{R}^{4\times 4}$	Random noise input superimposed on T_{init}
T_{init}	$\mathbb{R}^{4\times 4}$	Initial guess extrinsic matrix input (including T_Δ)
T_{pred}	$\mathbb{R}^{4\times 4}$	Prediction extrinsic matrix as a correction to T_{init}

encoded LiDAR input P to a voxel feature F_L^{3D}. These features are then fused into BEV features $F_\mathcal{B}$, which is subsequently decoded by GGBD component to get the prediction T_{pred}. In the training and evaluation process, the initial extrinsic matrix is constructed by superimposing a random noise T_Δ on top of the groundtruth T_{gt}. Hence, $T_{init} = T_\Delta \cdot T_{gt}$ (see more details in §3.2). Since T_Δ represents the random noise, a larger T_Δ means T_{init} will have a larger misalignment and make the problem more challenging. In our setting, we consider various magnitudes of perturbation up to $\{\pm 1.5m, \pm 20°\}$ as the noise range, representing a realistic and challenging calibration scenario. For evaluation, BEVCALIB takes I, P, K, and T_{init} as input, output a prediction T_{pred} to compensate for the injected noise. The final LiDAR to camera extrinsic prediction is $\hat{T}_{gt} = T_{pred}^{-1} \cdot T_{init}$. This strategy is useful to control the difficulty of the calibration problem without label leakage.

3.2 BEV Feature Extraction

BEV feature has an inherent geometric meaning, as each feature in BEV space corresponds to a specific area in the real world. In our setting, we use the LiDAR's coordinate as the world coordinate, which also serves as the BEV coordinate. Inspired by the previous cross-modal approaches [18], we adopt a similar paradigm that processes each modality separately and fuses them into a unified BEV feature space. Specifically, the LiDAR branch processes the input point cloud P using sparse convolutional backbone to produce a voxel feature $F_L^{3D} \in \mathbb{R}^{N_L\times X\times Y\times Z}$, which is then flattened to BEV features $\mathcal{B}_L^{2D} \in \mathbb{R}^{(N_L\times Z)\times X\times Y}$, where X, Y are the spatial shape of BEV plane, and Z is the number of vertical voxels along the height axis. The image branch leverages a 2D backbone and an LSS [42] module. The model first extracts the image feature $F_C^{2D} \in \mathbb{R}^{f_H\times f_W\times N_C}$ from camera input I, where f_H, f_W are the shape of image feature. The LSS module defines a discrete depth set for each pixel (u, v), termed as $\mathcal{D} = \{d_{min} + \frac{d_{max}-d_{min}}{D-1} \times i\}_{i=0}^{D-1}$, where D is the number of discrete depth bins. For each pixel (u, v), LSS produces D points, accumulating a frustum with $f_H \times f_W \times D$ points in total. The corresponding 3D features are represented as $F_C^{3D} \in \mathbb{R}^{D\times f_H\times f_W\times N_C}$, and the 3D positions in the camera coordinate is defined as $P_C \in \mathbb{R}^{D\times f_H\times f_W\times 3}$. To give the model an initial guess of the position, the frustum coordinates are transformed into world coordinates by $P_C^W = [T_{init}^{-1} \cdot \tilde{P}_C]_{1:3}$. Finally, we can get the camera's BEV features $\mathcal{B}_C^{2D} \in \mathbb{R}^{N_C\times X\times Y}$ using BEV pooling [2]. To get a unified BEV representation, we use a 1×1 convolution to fuse features from different modalities, *i.e.,* $F_\mathcal{B} = \text{Conv1D}([\mathcal{B}_C^{2D}, \mathcal{B}_L^{2D}]) \in \mathbb{R}^{N_\mathcal{B}\times X\times Y}$. We then adopt an FPN BEV Encoder to enhance the multi-scale geometric information of BEV representation.

3.3 Geometry-Guided BEV Decoder (GGBD)

Based on the geometric BEV representation of the scene, we further propose a Geometry-Guided BEV feature Decoder to learn meaningful geometry relationships between the camera and the LiDAR. As illustrated in Figure 2, the decoder consists of two stages: a feature selector and a refinement module. The BEV feature selector guides the model to focus on the BEV features with meaningful spatial information, while the refinement module aggregates high-level features and helps to predict the final extrinsic parameters.

Geometry-Guided BEV Feature Selector. Specifically for the feature selector, following the image branch of BEV feature extraction, we take the 3D feature positions P_C^W as anchors for cross-modal interaction by projecting them into BEV space. Specifically, for a 3D position $p_c = (x, y, z) \in P_C^W$, its corresponding BEV space coordinate is calculated by $x_B = \frac{X}{2} + \lfloor \frac{x}{s} \rfloor$, $y_B = \frac{Y}{2} + \lfloor \frac{y}{s} \rfloor$ where s is the size of the resolution of BEV's grids. We define the projection operation as $\text{Proj}(p) = (x_B, y_B)$, the set of selected BEV feature positions can be formulated as $P_\mathcal{B} = \text{Set}(\{\text{Proj}(p)|p \in$

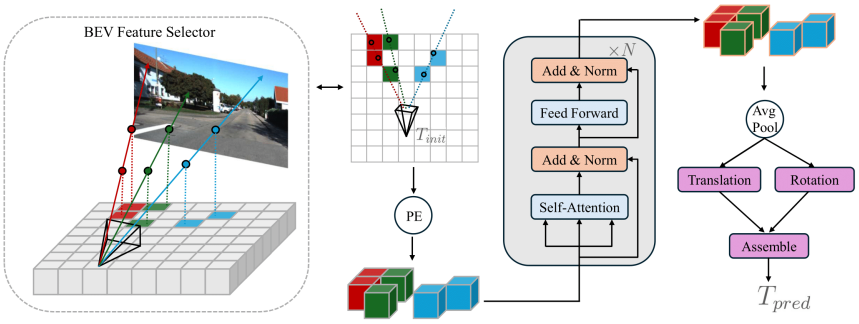


Figure 2: **Overall Architecture of Geometry-Guided BEV Decoder (GGBD).** The GGBD component contains a feature selector (left) and a refinement module (right). The feature selector calculates the positions of BEV features. The corresponding positional embeddings (PE) are added to keep the geometry information of the selected feature. The refinement module aggregates high-level information, following two separate heads to predict translation and rotation parameters.

$P_C^W\}$). Since the BEV space is a unified fused space shared by different modalities, such projection positions $(x_B, y_B) \in P_{\mathcal{B}}$ naturally provide a strong spatial prior for different modalities. This strategy inherently focuses on the overlapping regions between the camera and the LiDAR, acting as an implicit geometric matcher while eliminating redundant features.

Refinement Module. To illustrate the strength and generalizability of our geometric selector, we only use vanilla self-attention [46] as our refinement module. The whole process of the Geometry-Guided BEV Decoder (GGBD) can be written as

$$\text{GGBD}(P_C^W, F_{\mathcal{B}}) = \text{Self-Attention}\left(\phi_Q(F_\delta), \phi_K(F_\delta), \phi_V(F_\delta)\right) \tag{1}$$

$$F_\delta = \{F_B[:, x_B, y_B] \mid (x_B, y_B) \in P_{\mathcal{B}}\} \tag{2}$$

After GGBD, we apply an average-pooling operation to aggregate the feature. Subsequently, two separate multilayer perceptrons (MLPs) are used to predict translation and rotation, respectively. Finally, the predicted components are assembled into the final prediction T_{pred}.

3.4 Calibration Optimization

BEVCALIB outputs a translation vector $t \in \mathbb{R}^3$ and a rotation quaternion $r \in \mathbb{R}^4$, the supervision $\hat{r}$ and $\hat{t}$ is derived from $\hat{T}_{pred} = T_{init} \cdot T_{gt}^{-1} = \begin{bmatrix} \text{Q2M}(\hat{r}) & \hat{t} \\ 0 & 1 \end{bmatrix}$, where $\text{Q2M}(\hat{r})$ denotes the rotation matrix converted from quaternion $\hat{r}$. To effectively optimize the extrinsic calibration, we design a set of loss functions that focus on *rotation-only, translation-only, and joint calibration*.

Rotation Loss $\mathcal{L}_R$. For rotation, we adopt a geodesic loss [47] based on quaternion distance $\mathcal{L}_{ang} = 2\text{arctan2}\left(||q_\Delta^{(1:3)}||_2, |q_\Delta^{(0)}|\right)$, where $q_\Delta = r \cdot \hat{r}^{-1}$ is the relative quaternion between r and $\hat{r}$, $||\cdot||_2$ is l_2 norm and $|\cdot|$ is the absolute value. We also utilize a normalization loss to restrict the predicted quaternion r to be a valid rotation *i.e.,* $\mathcal{L}_{norm} = (||r||_2 - 1)^2$. Finally, $\mathcal{L}_R = \mathcal{L}_{ang} + \lambda_{norm}\mathcal{L}_{norm}$.

Translation Loss $\mathcal{L}_T$. For translation optimization, we use a Smooth-L1 loss to optimize it. We find that this loss alone is sufficient to optimize translation effectively, therefore, we don't incorporate additional objectives. The translational loss follows $\mathcal{L}_T = \text{Smooth-L1}\left(t, \hat{t}\right)$.

Reprojection Loss $\mathcal{L}_{PC}$. We use the point cloud reprojection loss introduced by LCCNet [9]. Specifically, it can directly supervise the alignment of the transformed point cloud using the predicted translation and rotation jointly, which can be written as $\mathcal{L}_{PC} = \frac{1}{N}\sum_{i=1}^{N} ||T_{gt}^{-1} \cdot T_{pred}^{-1} \cdot T_{init} \cdot \tilde{P}_i - \tilde{P}_i||_2$, where N is the number of points in the given point cloud P.

Total Loss Function. In summary, the combined loss function is $\mathcal{L} = \lambda_R\mathcal{L}_R + \lambda_T\mathcal{L}_T + \lambda_{PC}\mathcal{L}_{PC}$.

Implementation Details. We utilize sparse convolution [43] as the backbone for LiDAR and adopt Swin-Transformer [48] combined with LSS [42] as the backbone for the camera. For indoor datasets, we constrain the environment range to a 9-meter radius, while for outdoor datasets, we extend the range to 90 meters. We use a weight vector of (1.0, 0.5, 0.5) for the ($\mathcal{L}_R$, $\mathcal{L}_T$, $\mathcal{L}_{PC}$) losses, respectively, throughout all training runs. We trained BEVCALIB using only a single NVIDIA RTX

Table 2: Comparing with Original Results from Literature on KITTI [7]

Noise (Trans. Rot.)	Method	Magnitude↓		Translation (cm) ↓			Rotation (°) ↓		
		E_t(cm)	E_R(°)	**X**	**Y**	**Z**	**Roll**	**Pitch**	**Yaw**
(±1.5m, ±20°)	LCCRAFT [12]	37.57	1.44	31.38	12.87	16.17	1.30	0.42	0.47
	LCCNet [9]	15.00	0.94	11.85 ± 5.78	5.17 ± 3.85	7.61 ± 6.49	0.17 ± 0.26	0.68 ± 0.58	0.59 ± 0.22
	Regnet [8]	10.68	0.50	7	7	4	0.36	0.25	0.24
	Fu *et al.* [49]	3.27	0.28	2.26 ± 0.51	2.02 ± 0.87	**1.24 ± 0.58**	0.10 ± 0.02	0.21 ± 0.04	0.16 ± 0.03
	BEVCALIB **(Ours)**	**2.35**	**0.08**	**1.79 ± 1.61**	**0.48 ± 0.49**	1.45 ± 2.75	**0.03 ± 0.07**	**0.06 ± 0.08**	**0.04 ± 0.03**
(±0.5m, ±5°)	Koide3 [10]	21.04	0.55	6.9 ± 5.6	12.2 ± 14.9	15.7 ± 9.7	0.36 ± 0.14	0.23 ± 0.12	0.35 ± 0.22
	CalibAnything [11]	9.80	0.30	5.6 ± 4.0	5.0 ± 4.4	6.3 ± 6.2	0.19 ± 0.16	0.17 ± 0.10	0.16 ± 0.13
	3DGS-Calib [39]	9.60	0.45	9.60 ± 2.1 (xyz together)			0.45 ± 0.2 (rpy together)		
	SOAC [35]	7.80	0.30	7.8 ± 3.5 (xyz together)			0.3 ± 0.2 (rpy together)		
	BEVCALIB **(Ours)**	**2.46**	**0.06**	**1.78 ± 1.54**	**0.29 ± 0.26**	**1.67 ± 3.43**	**0.01 ± 0.07**	**0.05 ± 0.07**	**0.03 ± 0.02**
(±0.25m, ±10°)	CalibFormer [50]	2.11	0.29	**1.10**	0.90	1.56	0.08	0.26	0.09
	BEVCALIB **(Ours)**	**1.79**	**0.06**	1.44 ± 1.23	**0.25 ± 0.21**	**1.04 ± 2.40**	**0.02 ± 0.07**	**0.05 ± 0.07**	**0.03 ± 0.02**
(±0.2m, ±20°)	CalibNet [45]	8.50	0.93	4.2	1.6	7.22	0.15	0.90	0.18
	BEVCALIB **(Ours)**	**1.72**	**0.07**	**1.42 ± 1.24**	**0.29 ± 0.27**	**0.94 ± 1.55**	**0.02 ± 0.06**	**0.06 ± 0.08**	**0.04 ± 0.03**

6000 Ada GPU with a batch size of 16 for 500 epochs on each dataset (§4). We applied the AdamW optimizer with a weight decay of $1e^{-4}$ and an initial learning rate of $5e^{-5}$, which is decayed by a factor of 0.5 using a StepLR scheduler.

4 Evaluation

Datasets. To reproduce and compare with existing approaches, we use two of the most popular outdoor benchmarks in the LiDAR-camera calibration literature, KITTI [7] and NuScenes [14]. The comparison can contextualize BEVCALIB with related work. In the meantime, we also collected our own heterogenous extrinsic indoor dataset CALIBDB. CALIBDB includes 1244 traces. Each trace contains 12 seconds of continuous frames of image, LiDAR point cloud, and their dynamic extrinsic data recorded at 10 Hz. Our results show that BEVCALIB generalizes well on CALIBDB while this diversity poses significant challenges for existing calibration methods. Across all three datasets, we randomly split each dataset into 8:2 for training and testing without any pre-processing.

Metrics. We evaluate the translation and rotation error magnitude and break them down along each axis. The translation error is calculated as the L1 norm between the prediction and the groundtruth $|t_{gt} - t_{pred}|$. For rotation, we calculate the difference between the rotation matrices of prediction (R_{pred}) and groundtruth (R_{gt}), *i.e.,* $R_{pred}R_{gt}^T$, and extract the Euler angles.

Baselines. We compare BEVCALIB with two sets of baseline results, *original results reported in the publications* and *reproduced results from open-source methods*. In the first set, we include methods in the literature which has a similar evaluation setup (*e.g.,* noise range) such that the results can be compared fairly. These baselines include Fu et al. [49], LCCRAFT [12], LCCNet [9], SOAC [35], 3DGS-Calib [39], and CalibFormer [50]. In the second set, we tried our best to exhaust all publicly available and reproducible methods, including CalibAnything [11], Koide3 [10], Regnet [8], and CalibNet [45]. We use the official sources of CalibAnything and Koide3, and the officially recommended implementations of Regnet and CalibNet[1].

Notably, LCCNet [9] and LCCRAFT [12] use an iterative refinement approach during inference. Their methods first take a random guess similar to ours, then perform multiple inference passes, with each iteration's output serving as input for the next, progressively refining the calibration parameters. In contrast, our model utilizes a one-stage methodology; therefore, for a fair comparison, we only compare to the single-pass results. Several works are excluded from our evaluation either because they are not reproducible or cannot be compared fairly due to methodology differences. For example, CalibDepth [51] does not report single-pass results. MDPCalib [34] employs hybrid approaches that need additional heavy computation.

Quantitative Results. Table 2 and Table 3 compare BEVCALIB with the originally reported results from the publications on KITTI and NuScenes datasets. Since existing models were trained and evaluated using different noise settings, we group them into different clusters and evaluate BEVCALIB under the same noise settings for fair comparison. On KITTI, BEVCALIB has only a few centimeter translation error, outperforming the best baselines by an average of 14.29% - 78.82%, and less than

[1] We refer to the recommended unofficial implementations for CalibNet and Regnet.

Table 3: Comparing with Original Results from Literature on NuScenes [14]

Noise (Trans. Rot.)	Method	Magnitude↓		Translation (cm) ↓			Rotation (°) ↓		
		E_t(cm)	E_R(°)	**X**	**Y**	**Z**	**Roll**	**Pitch**	**Yaw**
(±0.5m, ±5°)	CalibAnything [11]	19.7	0.41	11.0 ± 7.4	10.0 ± 5.5	13.0 ± 12.2	0.2 ± 0.1	0.3 ± 0.2	**0.2 ± 0.1**
	Koide3 [10]	26.7	0.75	16.5 ± 12.1	15.6 ± 13.8	14 ± 11.9	0.5 ± 0.2	0.4 ± 0.3	0.4 ± 0.2
	BEVCalib (Ours)	**4.3**	**0.13**	**1.2 ± 0.7**	**6.7 ± 4.1**	**2.4 ± 2.4**	**0.2 ± 0.1**	**0.1 ± 0.1**	**0.2 ± 0.1**

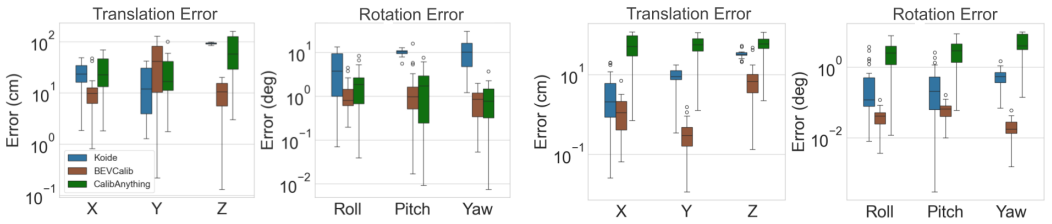
Table 4: Evaluation Results with Reproducible Open-source Baselines

Dataset	Method	Magnitude↓		Translation (cm) ↓			Rotation (°) ↓		
		E_t(cm)	E_R(°)	**X**	**Y**	**Z**	**Roll**	**Pitch**	**Yaw**
KITTI [7]	Regnet [8]	145.4	18.7	101.2 ± 0.9	67.8 ± 1.2	79.4 ± 0.8	16.3 ± 0.1	9.2 ± 0.0	0.8 ± 0.3
	CalibNet [45]	33.1	163.7	4.0 ± 3.8	6.5 ± 4.0	32.2 ± 5.9	98.4 ± 49.9	85.9 ± 2.9	98.7 ± 51.3
	CalibAnything [11]	101.1	6.8	59.0 ± 38.0	56.0 ± 30.2	60.0 ± 33.7	2.9 ± 2.1	3.2 ± 2.4	5.2 ± 3.0
	Koide3 [10]	35.4	0.9	4.7 ± 6.0	9.7 ± 3.9	33.7 ± 6.4	0.5 ± 0.9	0.5 ± 0.6	0.6 ± 0.3
	BEVCalib (Ours)	**2.4**	**0.1**	**1.8 ± 1.6**	**0.5 ± 0.5**	**1.5 ± 2.9**	**0.0 ± 0.1**	**0.1 ± 0.1**	**0.0 ± 0.1**
NuScenes [14]	Regnet [8]	196.1	93.6	95.2 ± 0.1	72.5 ± 0.4	155.3 ± 0.1	34.4 ± 0.6	71.5 ± 0.2	49.6 ± 0.4
	CalibNet [45]	83.6	87.5	5.1 ± 4.5	32.0 ± 7.4	77.1 ± 5.9	87.4 ± 3.4	2.1 ± 1.9	3.0 ± 2.6
	CalibAnything [11]	89.7	4.7	58.3 ± 28.3	51.2 ± 25.4	45.0 ± 30.3	2.1 ± 2.1	3.5 ± 2.8	2.4 ± 2.3
	Koide3 [10]	82.1	149.4	2.2 ± 1.8	32.3 ± 2.2	75.5 ± 1.8	85.2 ± 38.5	88.6 ± 0.6	84.9 ± 38.8
	BEVCalib (Ours)	**6.0**	**0.3**	**1.3 ± 1.0**	**5.4 ± 4.5**	**2.3 ± 2.4**	**0.2 ± 0.1**	**0.2 ± 0.1**	**0.2 ± 0.2**
CALIBDB	Regnet [8]	216.4	24.1	93.0 ± 27.1	43.2 ± 13.9	190.6 ± 5.8	17.6 ± 5.3	2.0 ± 1.3	16.4 ± 9.3
	CalibNet [45]	95.5	180.4	24.7 ± 14.1	17.4 ± 13.7	90.6 ± 9.2	72.6 ± 31.6	77.3 ± 4.1	145.9 ± 27.5
	CalibAnything [11]	86.2	3.3	31.0 ± 22.8	28.4 ± 25.0	75.3 ± 54.2	2.3 ± 2.2	2.1 ± 2.0	**1.0 ± 0.9**
	Koide3 [10]	96.8	16.5	24.0 ± 13.1	**17.3 ± 14.4**	92.2 ± 4.1	5.3 ± 4.9	10.2 ± 1.7	11.9 ± 9.2
	BEVCalib (Ours)	**38.0**	**2.5**	**8.4 ± 11.0**	36.4 ± 31.6	**6.9 ± 6.3**	**1.2 ± 1.2**	**1.7 ± 2.9**	1.3 ± 1.5

$0.1°$ rotation error, outperforming the best baselines by an average of 71.43% - 95.70% under various noise conditions. On Nuscenes, BEVCALIB has slightly bigger error but still outperforms the best baseline by 78.17% in translation, 68.29% in rotation. Notably, although BEVCALIB is trained under the largest noise (±1.5m, ±20°), it shows extreme robustness when evaluated on smaller noise, overcoming the noise sensitivity that cripples previous methods such as LCCNet [9]. In addition, BEVCALIB demonstrates remarkable rotation prediction accuracy for all three angles (roll, pitch, yaw) with error below 0.2°, achieving a near-perfect result that outperforms any previous methods.

Table 4 compares BEVCALIB with the reproducible baselines on KITTI, NuScenes, and CALIBDB. In our exhaustive effort searching for reproducible baselines, we find that the open-source space in this LiDAR-camera calibration domain is rather scarce (very few checkpoints) and underperforming despite the abundant literature. Hence, our open-source effort will significantly improve the performance of publicly available calibration tools. Specifically, Table 4 shows that BEVCALIB outperforms the best open-source baselines by (92.75%, 89.22%) on KITTI dataset and by (92.69%, 93.62%) on NuScenes dataset, in terms of (translation, rotation), respectively. While BEVCALIB approaches near-zero error on most, if not all, samples, CalibNet and Koide3 struggle with predicting the correct z-component while Regnet and CalibAnything struggle with all components on KITTI and NuScenes datasets. Across the board, when an initial guess is required, a random noise between $[-1.5m, 1.5m]$ and $[-20°, 20°]$ has been applied.

On our internal dataset CALIBDB, BEVCALIB still outperforms the best open-source baselines by (60.21%, 24.99%). Compared to KITTI and NuScenes, the error slightly increased for both translation and rotation. This can be attributed to the inherent difficulty of the heterogeneous extrinsics collected in CALIBDB. This characteristic is further illustrated in the error distribution shown in Figure 3. Compared to the error distribution when evaluating on KITTI, there is a larger gap between BEVCALIB and the baselines evaluated on CALIBDB.



(a) CALIBDB (b) KITTI

Figure 3: Error Distribution of BEVCALIB and Other Baselines on CALIBDB and KITTI.

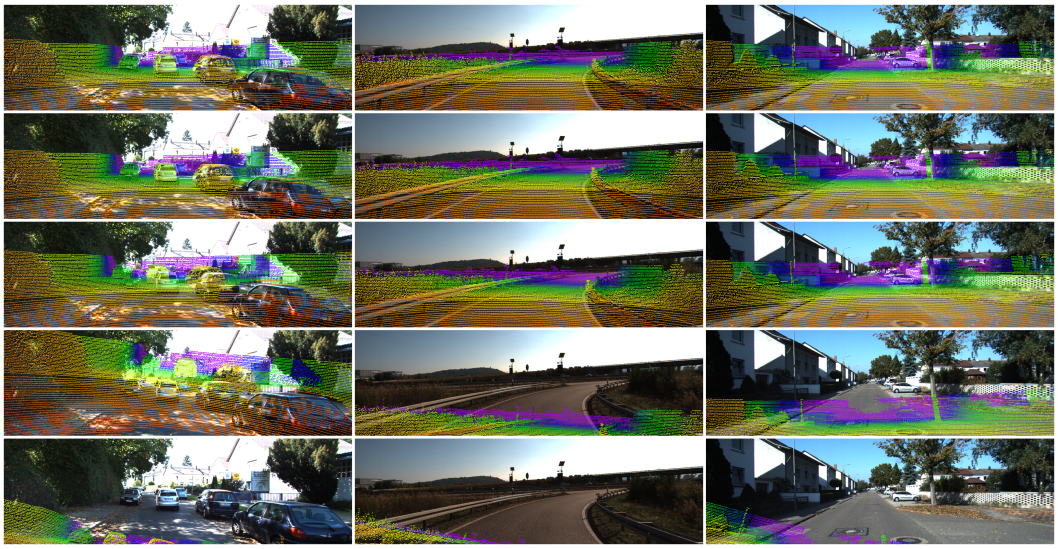


Figure 4: Qualitative results. A comparison of LiDAR-camera overlays from KITTI sequences. From top to bottom: ground-truth, BEVCALIB, Koide3 [10], CalibAnything [11], Regnet [8].

Table 5: Ablation Results

Method	Translation (cm) ↓			Rotation (°) ↓		
	X	Y	Z	Roll	Pitch	Yaw
BEVCALIB ∗	**8.4** ± **11.0**	**36.4** ± **31.6**	**6.9** ± **6.3**	**1.2** ± **1.2**	**1.7** ± **2.9**	**1.3** ± **1.5**
∗− BEV selector (use all features)	23.1 ± 19.5	37.5 ± 32.6	32.4 ± 17.6	2.5 ± 2.4	5.0± 3.9	2.7 ± 2.4
∗ with Deformable Attention	37.0 ± 30.8	37.0 ± 31.9	34.2 ± 27.3	5.6 ± 4.8	5.3 ± 4.3	5.3 ± 4.5

Qualitative Results. Figure 4 presents a qualitative comparison by overlaying the LiDAR point clouds over the image given each method's predicted extrinsic. Regnet and CalibAnything's overlays are misaligned due to the large error in rotation and translation, so the point cloud is not level with the ground. BEVCALIB and Koide3 are closer to the ground-truth overlay, but there are objects where Koide3's overlay is slightly misaligned, *e.g.,* the misaligned cars in the left column, the traffic sign in the middle column, and the pole and tree in the right column. In contrast, BEVCALIB's overlays do not show these misalignments. Overall, the overlays reflect the results in Table 4.

Runtime Performance. We evaluate BEVCalib on an NVIDIA RTX 4090, averaging 21.54, 41.84, and 30.69 FPS on CalibDB, KITTI, and NuScenes, respectively, with 0.80–1.13 GB of VRAM. Variations in FPS and peak VRAM reflect differing point-cloud densities across datasets.

Abalation Study. We first conduct an ablation to show the efficacy of the Geometry-Guided BEV feature selector in calibration optimization, where GGBD consists of a BEV selector and a refinement module. We investigate how different BEV feature selection strategies affect the refinement module. Table 5 shows that using all BEV features introduces too much redundant information to the model, significantly confusing the model about the cross-modality feature correspondence. We also experimented using different attention modules, *e.g.,* deformable attention [52], to capture the relationship between Camera and LiDAR, but the results are less ideal.

5 Conclusion

In this paper, we introduce BEVCALIB, the first LiDAR-camera extrinsic calibration model using BEV features. Geometry-guided BEV decoder can effectively and efficiently capture scene geometry, enhancing calibration accuracy. Results on KITTI, NuScenes, and our own indoor dataset with dynamic extrinsics illustrate that our approach establishes a new state of the art in learning-based calibration methods. Under various noise conditions, BEVCALIB outperforms the *best baseline in literature by an average of (47.08%, 82.32%) on KITTI dataset, and (78.17%, 68.29%) on NuScenes dataset*, in terms of (translation, rotation) respectively. Also, BEVCALIB improves the *best reproducible baseline by one order of magnitude*, making an important contribution to the scarce open-source space in LiDAR-camera calibration.

Limitations

Although BEVCALIB has demonstrated its exciting performance on calibration tasks for outdoor scenes, it still encounters difficulties when handling complex indoor scenes such as those in CALIBDB. Furthermore, BEVCALIB doesn't incorporate any pre-processing to mitigate dynamic elements; an explicit pre-processing pipeline could be expected to improve the model's robustness when facing dynamic scenes. Finally, while BEVCALIB successfully introduces different types of feature representation(BEV) to the calibration task, incorporating z-axis-explicit and depth-explicit representations such as volumetric and Tri-plane representations could further enhance its capabilities. We keep those limitations as our future directions to explore.

References

[1] A. J. Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha. Densecavoid: Real-time navigation in dense crowds using anticipatory behaviors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11345–11352, 2020. doi:10.1109/ICRA40945.2020.9197379.

[2] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[3] S. R. Mhatre and J. W. Bakal. Deepfusion: A novel deep learning technique for enhanced image super-resolution. In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pages 991–998, 2024. doi:10.1109/ICACRS62842.2024.10841630.

[4] J.-K. Huang and J. W. Grizzle. Improvements to Target-Based 3D LiDAR to Camera Calibration. *IEEE Access*, 8:134101–134110, 2020. doi:10.1109/ACCESS.2020.3010734.

[5] Q. 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, pages 2301–2306 vol.3, 2004. doi:10.1109/IROS.2004.1389752.

[6] G. Yan, F. He, C. Shi, P. Wei, X. Cai, and Y. Li. Joint camera intrinsic and lidar-camera extrinsic calibration. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11446–11452, 2023. doi:10.1109/ICRA48891.2023.10160542.

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[8] N. Schneider, F. Piewak, C. Stiller, and U. Franke. Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1803–1810, 2017. doi:10.1109/IVS.2017.7995968.

[9] X. Lv, B. Wang, Z. Dou, D. Ye, and S. Wang. Lccnet: Lidar and camera self-calibration using cost volume network. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2888–2895, 2021. doi:10.1109/CVPRW53098.2021.00324.

[10] K. Koide, S. Oishi, M. Yokozuka, and A. Banno. General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11301–11307. IEEE, 2023.

[11] Z. Luo, G. Yan, X. Cai, and B. Shi. Zero-training lidar-camera extrinsic calibration method using segment anything model. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14472–14478, 2024. doi:10.1109/ICRA57147.2024.10610983.

[12] Y.-C. Lee and K.-W. 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)*, pages 16669–16675, 2024. doi:10.1109/ICRA57147.2024.10610756.

[13] Q. Herau, N. Piasco, M. Bennehar, L. Roldão, D. Tsishkou, C. Migniot, P. Vasseur, and C. Demonceaux. Moisst: Multimodal optimization of implicit scene for spatiotemporal calibration. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 1810–1817. IEEE, Oct. 2023. doi:10.1109/iros55552.2023.10342427. URL http://dx.doi.org/10.1109/IROS55552.2023.10342427.

[14] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020. doi:10.1109/CVPR42600.2020.01164.

[15] P. Xiao, Z. Shao, S. Hao, Z. Zhang, X. Chai, J. Jiao, Z. Li, J. Wu, K. Sun, K. Jiang, Y. Wang, and D. Yang. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101, 2021. doi:10.1109/ITSC48978.2021.9565009.

[16] J. Shi, Z. Zhu, J. Zhang, R. Liu, Z. Wang, S. Chen, and H. 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)*, pages 10197–10202, 2020. doi:10.1109/IROS45743.2020.9341147.

[17] Y. Xiao, Y. Li, C. Meng, X. Li, J. Ji, and Y. Zhang. Calibformer: A transformer-based automatic lidar-camera calibration network, 2024. URL https://arxiv.org/abs/2311.15241.

[18] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022.

[19] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, , and J. M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *The Conference on Robot Learning (CoRL)*, 2021.

[20] H. Liu, Y. Teng, T. Lu, H. Wang, and L. Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos, 2023. URL https://arxiv.org/abs/2308.09244.

[21] Q. Li, Y. Wang, Y. Wang, and H. Zhao. Hdmapnet: An online hd map construction and evaluation framework. *arXiv preprint arXiv:2107.06307*, 2021.

[22] S. Choi, J. Kim, H. Shin, and J. W. Choi. Mask2map: Vectorized hd map construction using bird's eye view segmentation masks. In *European Conference on Computer Vision*, 2024.

[23] J. Ross, O. Mendez, A. Saha, M. Johnson, and R. Bowden. Bev-slam: Building a globally-consistent world map using monocular vision. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3830–3836, 2022. doi:10.1109/IROS47612.2022.9981258.

[24] L. Luo, S. Zheng, Y. Li, Y. Fan, B. Yu, S.-Y. Cao, J. Li, and H.-L. Shen. Bevplace: Learning lidar-based place recognition using bird's eye view images. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8666–8675, 2023. doi:10.1109/ICCV51070.2023.00799.

[25] Y. Zhang, Z. Zhu, and D. Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2304.05316*, 2023.

[26] J. Li, X. He, C. Zhou, X. Cheng, Y. Wen, and D. Zhang. Viewformer: Exploring spatiotemporal modeling for multi-view 3d occupancy perception via view-guided transformers. *arXiv preprint arXiv:2405.04299*, 2024.

[27] L. Zhang, Y. Xiong, Z. Yang, S. Casas, R. Hu, and R. Urtasun. Learning unsupervised world models for autonomous driving via discrete diffusion. *ICLR*, 2024.

[28] Y. Zhang, S. Gong, K. Xiong, X. Ye, X. Tan, F. Wang, J. Huang, H. Wu, and H. Wang. Bevworld: A multimodal world model for autonomous driving via unified bev latent space, 2024. URL https://arxiv.org/abs/2407.05679.

[29] S. Verma, J. S. Berrio, S. Worrall, and E. Nebot. Automatic extrinsic calibration between a camera and a 3d lidar using 3d point and plane correspondences. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3906–3912, 2019. doi:10.1109/ITSC.2019.8917108.

[30] R. Ishikawa, T. Oishi, and K. Ikeuchi. Lidar and camera calibration using motion estimated by sensor fusion odometry, 2018. URL https://arxiv.org/abs/1804.05178.

[31] G. Pandey, J. R. McBride, S. Savarese, and R. 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, page 2053–2059. AAAI Press, 2012.

[32] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946, 2020. doi:10.1109/CVPR42600.2020.00499.

[33] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, 2023. doi:10.1109/ICCV51070.2023.00371.

[34] K. Petek, N. Vödisch, J. Meyer, D. Cattaneo, A. Valada, and W. Burgard. Automatic targetless camera-lidar calibration from motion and deep point correspondences. *IEEE Robotics and Automation Letters*, 9(11):9978–9985, 2024.

[35] Q. Herau, N. Piasco, M. Bennehar, L. Roldao, D. Tsishkou, C. Migniot, P. Vasseur, and C. 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)*, pages 15131–15140, 2024. doi:10.1109/CVPR52733.2024.01433.

[36] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, Dec. 2021. ISSN 0001-0782. doi:10.1145/3503250. URL https://doi.org/10.1145/3503250.

[37] Z. Yang, G. Chen, H. Zhang, K. Ta, I. A. Bârsan, D. Murphy, S. Manivasagam, and R. Urtasun. Unical: Unified neural sensor calibration. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXVI*, page 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.

[38] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[39] Q. Herau, M. Bennehar, A. Moreau, N. Piasco, L. Roldao, D. Tsishkou, C. Migniot, P. Vasseur, and C. Demonceaux. 3dgs-calib: 3d gaussian splatting for multimodal spatiotemporal calibration, 2024.

[40] H. Li, C. Sima, J. Dai, W. Wang, L. Lu, H. Wang, J. Zeng, Z. Li, J. Yang, H. Deng, H. Tian, E. Xie, J. Xie, L. Chen, T. Li, Y. Li, Y. Gao, X. Jia, S. Liu, J. Shi, D. Lin, and Y. Qiao. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2023. doi:10.1109/TPAMI.2023.3333838.

[41] Y. Ma, T. Wang, X. Bai, H. Yang, Y. Hou, Y. Wang, Y. Qiao, R. Yang, and X. Zhu. Vision-centric bev perception: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10978–10997, 2024. doi:10.1109/TPAMI.2024.3449912.

[42] J. Philion and S. Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision*, 2020.

[43] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. ISSN 1424-8220. doi:10.3390/s18103337. URL https://www.mdpi.com/1424-8220/18/10/3337.

[44] W. Liao, S. Qiang, X. Li, X. Chen, H. Wang, Y. Liang, J. Yan, T. He, and P. Peng. Calibr-bev: Multi-camera calibration via reversed bird's-eye-view representations for autonomous driving. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 9145–9154, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706868. doi:10.1145/3664647.3680572. URL https://doi.org/10.1145/3664647.3680572.

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

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[47] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 2938–2946, USA, 2015. IEEE Computer Society. ISBN 9781467383912. doi:10.1109/ICCV.2015.336. URL https://doi.org/10.1109/ICCV.2015.336.

[48] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. URL https://arxiv.org/abs/2103.14030.

[49] L. F. T. Fu and M. F. Fallon. Batch differentiable pose refinement for in-the-wild camera/lidar extrinsic calibration. In *CoRL*, pages 1362–1377, 2023. URL https://proceedings.mlr.press/v229/fu23a.html.

[50] Y. Xiao, Y. Li, C. Meng, X. Li, J. Ji, and Y. Zhang. Calibformer: A transformer-based automatic lidar-camera calibration network. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16714–16720, 2024. doi:10.1109/ICRA57147.2024.10610018.

[51] J. Zhu, J. Xue, and P. Zhang. Calibdepth: Unifying depth map representation for iterative lidar-camera online calibration. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 726–733, 2023. doi:10.1109/ICRA48891.2023.10161575.

[52] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4794–4803, June 2022.