This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LiREC-Net: A Target-Free and Learning-Based Network for LiDAR, RGB, and Event Calibration

Aditya Ranjan Dash[1] Ramy Battrawy[2] René Schuster[1,2] Didier Stricker[1,2]
[1]RPTU – University Kaiserslautern-Landau
[2]DFKI – German Research Center for Artificial Intelligence
firstname.lastname@dfki.de

Abstract

*Advanced autonomous systems rely on multi-sensor fusion for safer and more robust perception. To enable effective fusion, calibrating directly from natural driving scenes (*i.e.*, target-free) with high accuracy is crucial for precise multi-sensor alignment. Existing learning-based calibration methods are typically designed for only a single pair of sensor modalities (*i.e.*, a bi-modal setup). Unlike these methods, we propose LiREC-Net, a target-free, learning-based calibration network that jointly calibrates multiple sensor modality pairs, including LiDAR, RGB, and event data, within a unified framework. To reduce redundant computation and improve efficiency, we introduce a shared LiDAR representation that leverages features from both its 3D nature and projected depth map, ensuring better consistency across modalities. Trained and evaluated on established datasets, such as KITTI and DSEC, our LiREC-Net achieves competitive performance to bi-modal models and sets a new strong baseline for the tri-modal use case.*

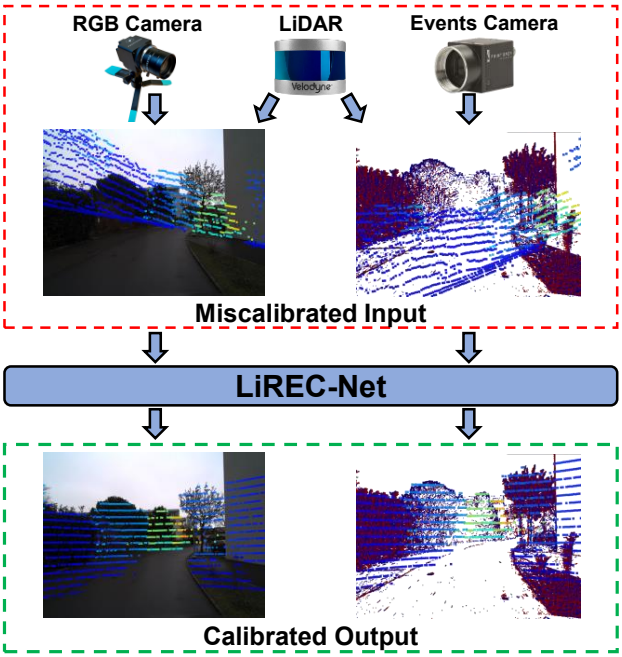


Figure 1. Our LiREC-Net takes miscalibrated tri-modal inputs and learns to produce spatially aligned outputs. The top row shows the raw miscalibrated overlays, while the bottom row illustrates the calibrated LiDAR projected onto both RGB and event frames.

1. Introduction

Modern vehicles with assisted and autonomous driving systems use multiple sensors together or fuse their outputs to build a consistent view of the environment. Calibration is needed to relate the relative poses of all sensors in a common coordinate system and to keep their measurements aligned in space. Sensors such as LiDAR, RGB cameras, and event cameras provide complementary information, and calibration ensures that this information corresponds spatially across views. In real deployments, vehicles experience vibration, temperature change, minor impacts, and routine maintenance, and over time, these factors can shift sensor poses even when the initial setup was correct. Traditional target-based methods, such as [4, 7, 17], which use checkerboards or markers, can be accurate, but they require controlled space, careful placement, repeated captures, and human oversight, which interrupts operation and is expensive to perform frequently. A target-free procedure avoids special setups and can be repeated whenever needed. Calibrating directly in natural scenes removes the need for controlled environments and reflects real operating conditions. Building on this idea, learning-based calibration methods leverage deep neural networks to extract and match cross-modal features from natural scenes. Methods such as LCCNet [16], RegNet [18], CalibNet [11], LCCRAFT [14], PseudoCal [2], and MULiEv [3] eliminate the need for calibration targets, improve robustness under varying conditions, and enable automated, target-free calibration. How-

ever, these methods are typically limited to only a single pair of sensor modalities (*i.e.*, the bi-modal setup), either LiDAR-RGB or LiDAR-Event, which duplicates effort and risks inconsistencies when extending to all three sensors. Inspired by these learning-based methods, we aim to address this gap by developing a unified, tri-modal calibration framework that jointly handles LiDAR, RGB, and event cameras. We propose LiREC-Net, an end-to-end learning-based calibration network that performs extrinsic calibration directly from natural driving scenes, as illustrated in Fig. 1, without requiring special targets or controlled environments. The same model can also operate in reduced pairwise configurations, while maintaining a consistent alignment path across all three modalities when available.

Our main contributions are:

- LiREC-Net – a unified and novel tri-modal neural network for calibrating LiDAR, RGB, and event cameras, improving both calibration accuracy and efficiency.
- A shared LiDAR representation used by both the LiDAR-RGB and LiDAR-Event paths, improving consistency across pairs and reducing redundancy.
- A point-cloud encoding strategy that fuses the 3D structure with their projected depth maps, enhancing the overall point cloud representation and improving accuracy.
- A strong tri-modal baseline with consistent calibration results across datasets and competitive errors compared to established bi-modal methods.

2. Related Work

Calibration has been extensively explored in pairwise (*i.e.*, bi-modal) settings, encompassing target-based, target-free, and learning-based approaches. Below, we review prior work on LiDAR-RGB and LiDAR-Event calibration.

2.1. LiDAR-RGB Calibration

Target-based approaches use multiple checkerboards to obtain sufficient constraints for full 6-DoF calibration [7]. Likewise, Dhall *et al*. [4] use ArUco markers for 3D–3D calibration with a fast closed-form extrinsic solution. Beyond target-based approaches, target-free methods operate in natural scenes by maximizing mutual information between image intensity and LiDAR reflectance [17].

Deep learning enables a new class of target-free calibration methods, in which a neural network is trained to align the sensors' data. RegNet [18] is one of the earliest models in this category, directly regressing extrinsic parameters from RGB images and 3D point clouds. CalibNet [11] introduces a geometric, self-supervised approach using depth-photometric and point-cloud distance losses. LCCNet [16] predicts extrinsics using a regression framework built on cost volumes constructed from image features and projected LiDAR-depth features, capturing cross-modal similarity. LCCRAFT [14] adapts RAFT [21] for calibration by iteratively updating the pose estimate, achieving improved robustness to large initial miscalibrations. PseudoCal [2], which uses PointPillars [13] for feature extraction, processes the RGB image with [23] to generate a 3D pseudo-LiDAR that is then used for calibration together with the real LiDAR. To further improve accuracy, Cocheteux *et al*. [1] employ two cascaded refinement stages, UniCal-M and UniCal-S, yielding precise extrinsic calibration.

Our work draws inspiration from LCCNet [16] and extends the core concept to a unified tri-modal setting.

2.2. LiDAR-Event Calibration

Similar to LiDAR-RGB calibration, LiDAR-Event calibration has been explored with conventional methods. For example, L2E [20] projects the LiDAR into a 2D image and recovers the pose by maximizing mutual information between projected LiDAR intensity and accumulated event frames. EF-Calib [22] jointly estimates intrinsics, extrinsics, and time offset for event and frame cameras using a custom calibration pattern and a continuous-time motion model. The method detects the shared pattern in both sensors, models motion with a piecewise B-spline, derives analytical Jacobians, and solves a spatiotemporal bundle-like optimization to recover all parameters jointly.

Target-free, learning-based LiDAR-Event calibration is scarcely explored, with MULiEv [3] being the only representative method. Using the DSEC dataset [5], MULiEv concatenates the LiDAR depth image and event frame channel-wise, extracts features with MViTV2 [15], and predicts the extrinsics. It also employs iterative refinement, similar to LCCNet [16], but with fewer stages.

In contrast to prior learning-based approaches that address only LiDAR-RGB or LiDAR-Event calibration independently, our method jointly calibrates both pairs within a single target-free architecture. It reuses a shared LiDAR backbone with point-based and depth-based feature fusion, constructs pairwise cost volumes for each modality, and supports operation with either or both camera inputs.

3. LiREC-Net

3.1. Problem Formulation

In this work, we address the joint calibration of a LiDAR sensor, an RGB camera, and an event camera. Each training or evaluation sample consists of a miscalibrated LiDAR point cloud $P \in \mathbb{R}^{N\times 3}$, an RGB image $I \in \mathbb{R}^{H\times W\times 3}$, an event representation $E \in \mathbb{R}^{H\times W\times 2}$, and known camera intrinsics $\mathbf{K}_{\mathrm{RGB}}$ and $\mathbf{K}_{\mathrm{Ev}}$. The network takes these three modalities as input and predicts the LiDAR-to-camera extrinsics (*i.e.*, relative poses) for multiple modality pairs $v \in \{\mathrm{Li}-\mathrm{RGB}, \mathrm{Li}-\mathrm{Ev}\}$. These extrinsics are parameterized by a translation $\hat{\mathbf{t}}^v \in \mathbb{R}^3$ and a quaternion rotation $\hat{\mathbf{q}}^v \in \mathbb{R}^4$, yielding the predicted transformations $\hat{\mathbf{T}}^v$. We

assume that the transformations per pair are different from each other, while RGB and event cameras are assumed to be pre-calibrated, so the relative pose $\mathbf{T}^{\mathrm{Ev}\to\mathrm{RGB}}$ is known. When only one camera modality is available, the model predicts the corresponding single transformation.

3.2. Architecture

LiREC-Net follows a dual-path design that jointly handles LiDAR-RGB and LiDAR-Event calibration within a unified framework. The overall architecture, shown in Fig. 2, consists of modality-specific feature encoders followed by pairwise cost volumes, context modules, and prediction heads.

3.2.1. Input Processing

As LiDAR, RGB, and event cameras are different modalities, we preprocess them separately before encoding. All three sensors need to be temporally synchronized before preprocessing. The following outlines the preprocessing steps for each.

LiDAR. For the LiDAR input, we first transform all points into a camera coordinate system, remove those with negative depth, and apply a cutoff at a maximum range $Z_{\max}$. The resulting point cloud is then resampled to a fixed number of N points. Frames with more than N points are randomly downsampled, while frames with fewer points are upsampled by duplicating randomly selected points. This processed set of points is used by two dedicated LiDAR encoders. A point-based encoder operates directly on the N three-dimensional points, while a parallel depth-based encoder projects the same points into an image plane to generate a single-channel depth map at the model's input resolution $H' \times W'$. This is achieved by scaling the camera intrinsics through multiplication with a scale matrix R'.

$$R' = \mathrm{diag}\big(\frac{W'}{W}, \frac{H'}{H}, 1\big), \quad \mathbf{K}'_{\mathrm{Cam}} = R'\,\mathbf{K}_{\mathrm{Cam}} \tag{1}$$

We call this projection *scaled depth projection* (SDP). It is essential to our architecture and proved impactful in our experiments (*cf*. Sec. 4.6). In comparison to projection with the original intrinsic followed by resizing of the depth map, SDP introduces fewer blurring artifacts, which are harmful for precise feature alignment.

RGB and Event Camera. For the RGB image input, we apply per-channel standardization using statistics of the training set by subtracting the channel means and dividing by the corresponding standard deviations. We resize the images to the model's input resolution ($H' \times W'$) using bilinear interpolation and feed them to the RGB feature encoder. For events, we accumulate the stream over a temporal window of 50ms to construct two-channel frames with positive and negative polarities. After time synchronization with the RGB and LiDAR clocks, we slice events in a window centered on each RGB timestamp. As stated in our prerequisite, the event and RGB cameras are assumed to be extrinsically calibrated. We apply the known extrinsics (*cf*. Sec. 3.1) to transform the event data into the RGB camera coordinate system, ensuring alignment between the event frame and the RGB image. Finally, we resize the two-channel event frame to the model's input resolution using bilinear interpolation.

3.2.2. Encoding

Shared LiDAR Branch. The shared LiDAR branch extracts complementary geometric information using two parallel encoders, one *point-based* and one *depth-based*, as shown in the green block of Fig. 2. The point-based encoder uses Point-Transformer-V3 (PTV3) [24], which processes unordered 3D points by serializing them through space-filling curves to enable efficient local attention and capture fine-grained geometric structure while remaining computationally scalable. The depth-based encoder employs Mobile-Vision-Transformer-V2 (MViTV2) [15] to capture spatial context from the projected depth maps.

The per-point features produced by the point-based branch are projected onto an image plane by scaling the camera intrinsics based on Eq. 1, ensuring consistency with the resolution of the depth features $H'' \times W''$. We refer to this process as *scaled feature projection* (SFP). Similar to SDP, SFP is beneficial for the same underlying reasons.

The projected point features and depth features are brought to a common channel dimension and fused through channel-wise concatenation, forming a unified LiDAR embedding that combines detailed 3D structure from point features and dense geometric cues from depth features. This fusion improves rotation and translation estimation, as confirmed by our experiments. The shared LiDAR branch benefits both the LiDAR-RGB and LiDAR-Event branches and results in improved efficiency with lower inference time, parameter count, and memory usage, compared to having separate LiDAR branches for each modality pair (see Tab. 3).

RGB and Event Encoders. The RGB and event branches extract visual features using two independent encoders based on Mobile-Vision-Transformer-V2 (MViTV2) [15]. MViTV2 is a hybrid architecture that combines convolutional layers with transformer blocks, enabling efficient modeling of local spatial patterns together with global context. The convolutional layers capture fine-grained textures and edges, while the transformer components aggregate information across the image to encode long-range dependencies. The RGB and event encoders process their respective preprocessed inputs, each independently learning modality-specific representations that capture appearance and motion cues critical for accurate geometric calibration.

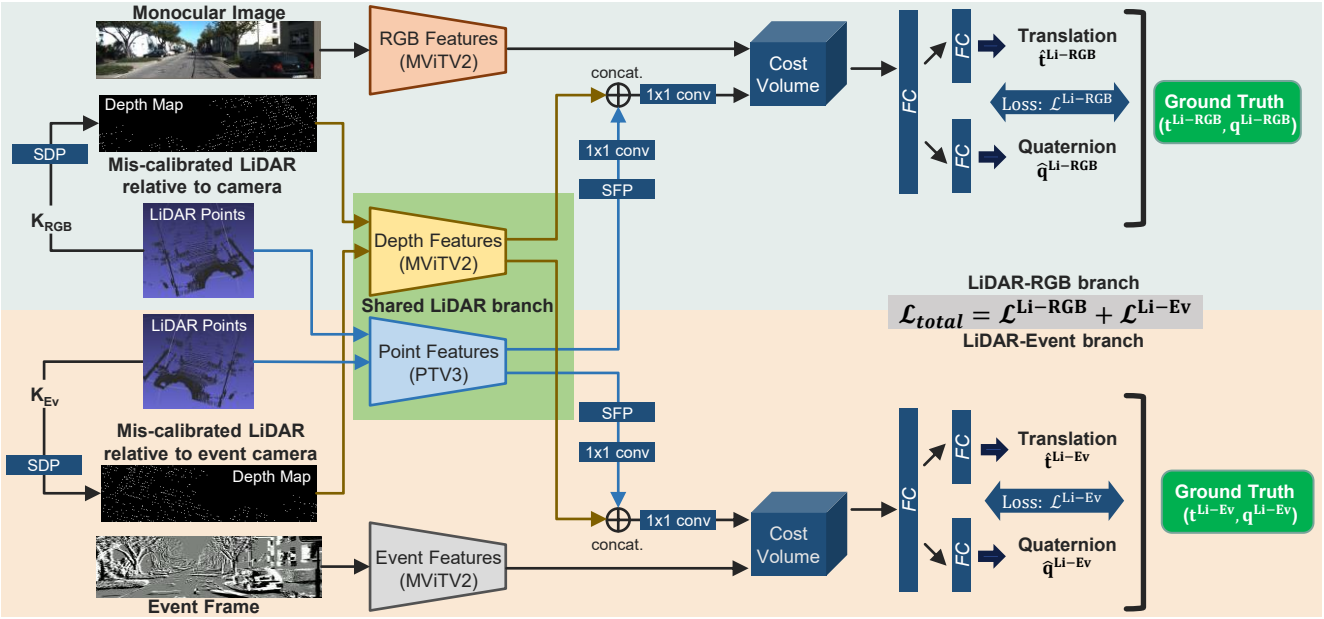


Figure 2. Overview of LiREC-Net. A miscalibrated LiDAR point cloud P is processed by two LiDAR encoders (point- and depth-based). Point features are projected to the image plane using known intrinsics $\mathbf{K}_{\text{RGB}}$ and $\mathbf{K}_{\text{Ev}}$, and then *fused* with depth features to form a unified LiDAR embedding. In parallel, the RGB image I and event representation E are encoded by their respective encoders. The unified LiDAR embedding is combined with the corresponding RGB/event features to build two pair-wise cost volumes, which are refined by context modules and passed to prediction heads that output the LiDAR-RGB and LiDAR-Event extrinsics $\hat{\mathbf{T}}^{\text{Li-RGB}}$ and $\hat{\mathbf{T}}^{\text{Li-Ev}}$.

3.2.3. Pair-wise Cost Volumes

After feature extraction, two parallel modality-specific paths (LiDAR-RGB and LiDAR-Event) construct correlation cost volumes to measure local cross-modal similarity, following PWC-Net [19] and LCCNet [16]. Let $\mathbf{F}^{\text{Li}}, \mathbf{F}^{Cam \in \{\text{RGB,Ev}\}} \in \mathbb{R}^{H'' \times W'' \times C}$ be the two feature maps of the shared LiDAR branch and one of the camera branches. For a pixel $\mathbf{p} = (x, y)$ and a displacement $(\Delta x, \Delta y) \in \{-d, \ldots, d\}$ within a square window of radius d, the matching cost is the channel-normalized inner product as defined in the following equation:

$$\mathcal{C}(y, x, \Delta x, \Delta y) = \frac{1}{C} \sum_{c=1}^{C} \mathbf{F}^{\text{Li}}_{c,y,x} \cdot \mathbf{F}^{\text{Cam}}_{c,\, y+\Delta y,\, x+\Delta x} \quad (2)$$

The costs over all displacements form a three-dimensional cost volume $\mathcal{C} \in \mathbb{R}^{H'' \times W'' \times M}$. Here, M is the number of offset locations in the square window $M = (2d+1)^2$. The cost volume computes the correlation between the two feature vectors for every pixel and offset. We then apply a LeakyReLU activation and refine the volume using a context module before predicting the extrinsics.

3.2.4. Context Modules

For the two modality-specific paths (LiDAR-RGB and LiDAR-Event), the context module has identical topologies and follows the design in LCCNet [16]. Each consists of five convolutional layers, and after each layer, the output features are concatenated with the running tensor, as in DenseNet [10]. This enhances the channel capacity and preserves early representation, resulting in richer pair-specific latent features. Concatenating features – rather than overwriting them – gradually expands the channel dimension while retaining both low- and high-level information. The resulting expressive representation is then passed to the fully connected prediction heads.

3.2.5. Prediction Heads

Similar to the pair-wise cost volume and pair-wise context modules, we use two identical calibration heads, one for the LiDAR-RGB path and one for the LiDAR-Event path. For each case, the resulting pair-specific latent features from the dense convolutional layers are flattened into a one-dimensional vector and passed through a shared fully connected layer. The resulting representation is then split into two fully connected prediction heads: one for translation and one for rotation. The translation head uses two fully connected layers with a LeakyReLU in between to regress the three-dimensional translation vector. The rotation head follows a similar design with two fully connected layers and a LeakyReLU in between, but its output is normalized to enforce a valid quaternion representation. This split prediction head design enables the network to learn translation and rotation simultaneously.

3.3. Iterative Refinement

Inspired by [16], we train multiple independent models, each specialized for different random translation and rotation errors within a specific range, starting from the largest to the smallest error range (*cf*. Sec. 4.2.1). During evaluation, each stage predicts a transformation that partially corrects the miscalibration, and the partially corrected result serves as input to the next stage. Let $k = 1, \ldots, S$ index the stages, let $\hat{\mathbf{T}}^{(0)}$ be the initial estimate, representing the identity transformation, *i.e*., the initial miscalibrated pose is not corrected, and let $\Delta\hat{\mathbf{T}}^{(k)}$ be the residual predicted by stage k. We update via:

$$\hat{\mathbf{T}}^{v,(k)} = \Delta\hat{\mathbf{T}}^{v,(k)}\,\hat{\mathbf{T}}^{v,(k-1)}, \qquad k = 1, \ldots, S \tag{3}$$

3.4. Loss Functions

The network is trained with three losses per modality pair: a translation loss, a rotation loss, and a point cloud distance loss. The translation and rotation loss directly supervise the predicted pose parameters, while the point-cloud distance loss enforces geometric alignment between the LiDAR points transformed by the predicted pose and the same points transformed by the ground truth pose.

Translation loss. We use a Smooth L1 loss on the predicted translation vector $\hat{\mathbf{t}}^v \in \mathbb{R}^3$ and the ground truth translation $\mathbf{t}^v \in \mathbb{R}^3$ over a batch of size B:

$$\mathcal{L}^v_{\text{trans}} = \frac{1}{B}\sum_{i=1}^{B} \|\hat{\mathbf{t}}^v_i - \mathbf{t}^v_i\|_1^{\text{smooth}} \tag{4}$$

Rotation loss. Rotations are supervised using the angular distance $\theta(\cdot,\cdot)$ between predicted unit quaternion $\hat{\mathbf{q}}^v$ and ground truth unit quaternion $\mathbf{q}^v$:

$$\mathcal{L}^v_{\text{rot}} = \frac{1}{B}\sum_{i=1}^{B} \theta(\hat{\mathbf{q}}^v_i, \mathbf{q}^v_i) \tag{5}$$

Point cloud distance loss. To ensure geometric consistency, we measure the alignment of LiDAR points $\mathbf{x} \in P_i$ after applying the predicted and ground truth rigid transformations $\hat{\mathbf{T}}^v$ and $\mathbf{T}^v$:

$$\mathcal{L}^v_{\text{pcd}} = \frac{1}{B}\sum_{i=1}^{B} \frac{1}{|P_i|}\sum_{\mathbf{x}\in P_i} \|\hat{\mathbf{T}}^v(\mathbf{x}) - \mathbf{T}^v(\mathbf{x})\|_2 \tag{6}$$

Total loss. The following formulation defines the per-pair loss used for the LiDAR-RGB and LiDAR-Event calibrations, where $\lambda_t, \lambda_r \geq 0$ control the translation and rotation contributions, and $w \in [0, 1]$ balances the point cloud loss:

$$\mathcal{L}^v = (1-w)(\lambda_t \mathcal{L}^v_{\text{trans}} + \lambda_r \mathcal{L}^v_{\text{rot}}) + w\mathcal{L}^v_{\text{pcd}} \tag{7}$$

The total training objective is obtained by summing the losses from both LiDAR-RGB and LiDAR-Event pairs:

$$\mathcal{L}_{\text{total}} = \mathcal{L}^{\text{Li-RGB}} + \mathcal{L}^{\text{Li-Ev}} \tag{8}$$

4. Experiments and Results

4.1. Datasets

To ensure a fair comparison with state-of-the-art calibration methods and to achieve our target of handling three sensing modalities, we use the KITTI Odometry dataset [6] and DSEC [5] datasets.

4.1.1. KITTI Dataset

The KITTI dataset [6] is widely used for LiDAR-RGB calibration and provides accurate, temporally synchronized data from a Velodyne HDL-64E LiDAR and Point Grey Flea stereo RGB cameras; we use the left RGB camera together with the LiDAR for all experiments. Since KITTI does not include event data, we generate synthetic events from the RGB frames using V2E [9]. For preprocessing, each LiDAR frame is limited to $N = 20{,}000$ front-facing points, based on the average number of front-facing points across the dataset.

4.1.2. DSEC Dataset

The DSEC dataset [5] provides all three sensing modalities required for our work, LiDAR, RGB, and events, captured by a Velodyne VLP-16 LiDAR, a FLIR Blackfly-S RGB stereo setup, and a Prophesee Gen3.1 event stereo setup, respectively. We use the left RGB camera, the left event camera, and the LiDAR for all experiments, and synchronize the three modalities using the provided timestamps. The LiDAR in DSEC is four times sparser than in KITTI, and the RGB images span diverse scenes with varying lighting and weather conditions, making the dataset more realistic and challenging. Each LiDAR frame is limited to $N = 5{,}000$ front-facing points to account for its lower density.

4.2. Implementation and Training Details

4.2.1. Artificial LiDAR Perturbation

To simulate miscalibration during training and evaluation, we apply artificial perturbations to the ground truth relative poses. The network learns to recover the correct extrinsics from these perturbed inputs. Two perturbation setups are used for comparability with prior works. Firstly, we adopt a five-stage range with maximum translation and rotation errors of $\pm\{20°/150\,\text{cm},\ 10°/100\,\text{cm},\ 5°/50\,\text{cm},\ 2°/20\,\text{cm},\ 1°/10\,\text{cm}\}$, to allow fair comparison with RegNet [18] and LCCNet [16]. Secondly, we use a two-stage range of $\pm\{10°/100\,\text{cm},\ 1°/10\,\text{cm}\}$, to allow fair comparison with MULiEv [3].

During training, the first stage is trained from scratch using the largest perturbations within the range, while each

subsequent stage is initialized from its predecessor and trained on inputs with progressively smaller perturbations. At each stage, perturbations are sampled uniformly from the corresponding range. During evaluation, we apply only one random perturbation from the largest stage, and then pass the input sequentially through all stages.

Importantly, two distinct perturbations are applied to the pair-wise modalities to simulate independent LiDAR-RGB and LiDAR-Event miscalibrations, ensuring that the two cameras are miscalibrated (*i.e.*, perturbed) differently with respect to the LiDAR during both training and testing (*cf.* independent LiDAR perturbations in Fig. 2). However, this does not resolve our assumption that the RGB and event cameras must be pre-calibrated.

4.2.2. Hyperparameters

We train LiREC-Net using the Adam optimizer [12] with a learning rate of 3×10^{-4}, which decays by a factor of 0.5 at milestones determined through manual experimentation based on validation loss saturation. For DSEC, the first stage is trained for 150 epochs, and each subsequent stage is initialized from the previous one and trained for 70 epochs. For KITTI, being a smaller dataset compared to DSEC, the first stage is trained for 120 epochs, and each subsequent stage is trained for 50 epochs. More details on the learning rate decay and all other hyperparameters are provided in our supplementary material. The model is trained with a batch size of 64 on four NVIDIA RTXA6000/L40S GPUs. For a fair comparison of efficiency, all models are evaluated on the same NVIDIA H200 GPU under identical settings.

4.3. Evaluation Metrics

We evaluate calibration accuracy using the mean translation and rotation errors computed over all test samples N_s between the predicted and ground truth transformations. The translation error e_t is computed as the mean Euclidean distance between the predicted translation vectors $\hat{\mathbf{t}}^v$ and the ground truth translation vectors $\mathbf{t}^v$ (see Eq. 9). The rotation error e_r is defined as the mean angular difference θ between the predicted quaternions $\hat{q}^v$ and the ground truth quaternions q^v (see Eq. 10).

$$\mathrm{e}_t = \frac{1}{N_s} \sum_{i=1}^{N_s} \|\hat{\mathbf{t}}_i^v - \mathbf{t}_i^v\|_2 \tag{9}$$

$$\mathrm{e}_r = \frac{1}{N_s} \sum_{i=1}^{N_s} \theta(\hat{q}_i^v, q_i^v) \tag{10}$$

4.4. Comparison to State-of-the-Art

We evaluate LiREC-Net against several state-of-the-art calibration methods on the KITTI and DSEC datasets, as summarized in Tabs. 1 and 2.

On KITTI, LiREC-Net achieves competitive calibration accuracy for LiDAR-RGB, outperforming CalibNet [11] and RegNet [18]. We surpass LCCNet [16] in terms of rotation, and achieve a marginally higher translation error with a difference of 21mm. PseudoCal [2] shows a translation difference of 62mm and a rotation difference of $0.06°$ (see Tab. 1). We furthermore outperform LCCRAFT [14] and present the comparison in the supplementary, since LCCRAFT uses different evaluation metrics. Considering both translation and rotation metrics, LiREC-Net achieves competitive overall performance with only marginal differences compared to existing methods. Notably, LiREC-Net achieves 1.80cm / $0.11°$ on LiDAR-RGB and 1.82cm / $0.12°$ on LiDAR-Event, highlighting its ability to calibrate multiple modality pairs. Furthermore, LiREC-Net is the first to address LiDAR-Event calibration on KITTI using synthetic events and therefore establishes the first baseline in this category.

On DSEC, where event data is more prevalent and captured with a real sensor, LiREC-Net achieves strong calibration accuracy across both LiDAR-RGB and LiDAR-Event pairs (see Tab. 2). For LiDAR-RGB, no prior work reports results on this dataset. Thus, our method provides the first benchmark with a translation and rotation error of 2.51cm and $0.14°$, establishing a new baseline for this modality pair. For LiDAR-Event, while MULiEv [3] achieves a slightly lower translation error (0.81cm *vs.* 1.18cm), LiREC-Net attains superior rotation accuracy ($0.07°$ *vs.* $0.10°$) and performs competitively overall, while additionally handling multiple modalities within a single framework. These results confirm LiREC-Net as the first unified solution for LiDAR-RGB and LiDAR-Event calibration on DSEC.

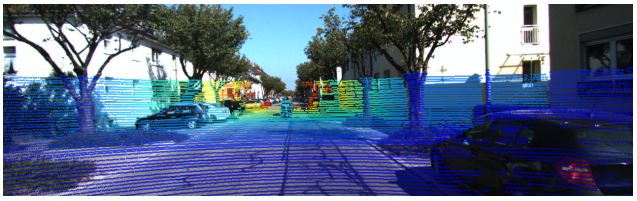
Table 1. Calibration results on the KITTI dataset when LiREC-Net is trained and evaluated with maximum perturbation of $\pm(20°/150\,\mathrm{cm})$.

Network	**LiDAR-RGB** e_t [cm] / e_r [°]	**LiDAR-Event** e_t [cm] / e_r [°]
RegNet [18]	6.00 / 0.28	— / —
CalibNet [11]	4.34 / 0.41	— / —
LCCNet [16]	1.59 / 0.16	— / —
PseudoCal [2]	**1.18 / 0.05**	— / —
LiREC-Net (Ours)	1.80 / 0.11	**1.82 / 0.12**

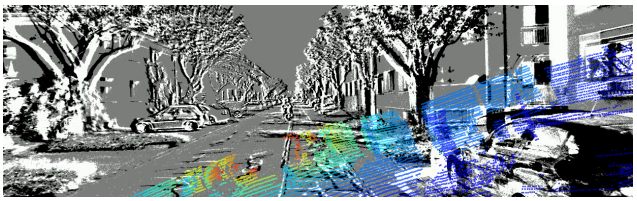
Table 2. Calibration results on the DSEC dataset. [*] Training/evaluation with 5 stages at $\pm(20°/150\,\mathrm{cm})$, [$^\triangle$] with 2 stages at $\pm(10°/100\,\mathrm{cm})$.

Network	**LiDAR-RGB** e_t [cm] / e_r [°]	**LiDAR-Event** e_t [cm] / e_r [°]
MULiEv$^\triangle$ [3]	— / —	**0.81** / 0.10
LiREC-Net$^\triangle$ (Ours)	2.62 / 0.30	2.05 / 0.25
LiREC-Net* (Ours)	**2.51 / 0.14**	1.18 / **0.07**

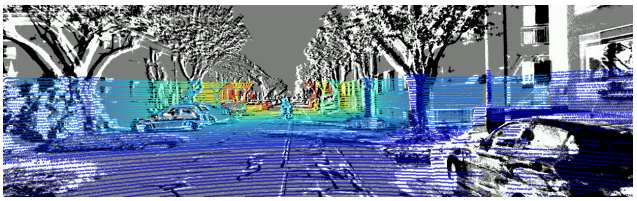
(a) LiDAR-RGB with perturbations within $\pm(20°/150\,\text{cm})$.



(b) LiDAR-RGB calibration using LiREC-Net's predicted transformation.



(c) LiDAR-Event with perturbations within $\pm(20°/150\,\text{cm})$.



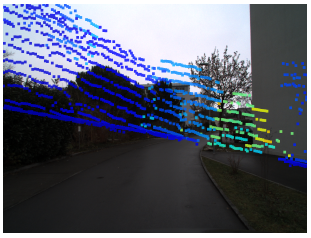
(d) LiDAR-Event calibration using LiREC-Net's predicted transformation.

Figure 3. Qualitative results on KITTI [6] for both LiDAR-RGB and LiDAR-Event pairs.

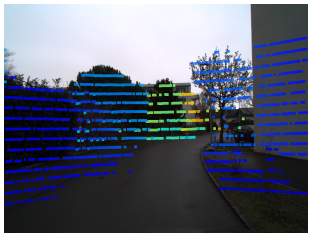
Figures 3 and 4 illustrate qualitative results on the KITTI and DSEC datasets for both LiDAR-RGB and LiDAR-Event pairs. The visualizations each show the projected LiDAR point cloud on top of the respective camera frame before and after calibration with LiREC-Net. Across both datasets and modalities, LiREC-Net effectively restores spatial alignment between LiDAR projections and the corresponding frames, demonstrating accurate and robust cross-modal calibration. More qualitative results are provided in our supplementary material.

4.5. Bi-modal vs. Tri-modal Calibration

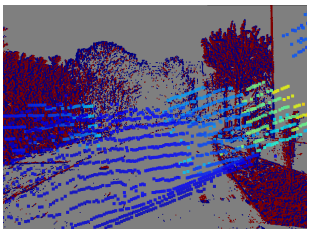
We analyze the impact of jointly training LiREC-Net on both LiDAR-RGB and LiDAR-Event (*i.e.*, *tri-modal*), compared to training separate networks for each modality pair (*i.e.*, *bi-modal*), where the latter refers to independent training and evaluation for each pair (*cf*. Tab. 3). For the bi-modal case, the reported inference time (T), parameter count (P), and GPU memory (M) are summed over the two separate models. For a fair comparison of efficiency, all models are evaluated using the experimental setup detailed in Sec. 4.2.2. The tri-modal setting demonstrates comparable or superior calibration accuracy with improved efficiency. This efficiency gain stems from the shared LiDAR branch, which avoids duplicated parameters and allows for more efficient computation without compromising on performance.



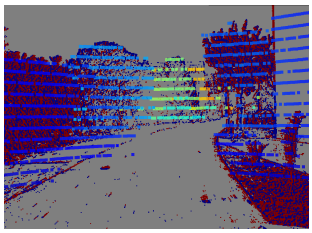
(a) LiDAR-RGB with perturbations within $\pm(20°/150\,\text{cm})$.



(b) LiDAR-RGB calibration using LiREC-Net's predicted transformation.



(c) LiDAR-Event with perturbations within $\pm(20°/150\,\text{cm})$.



(d) LiDAR-Event calibration using LiREC-Net's predicted transformation.

Figure 4. Qualitative results on DSEC [5] for both LiDAR-RGB and LiDAR-Event pairs.

Table 3. Calibration result comparison when LiREC-Net is trained and evaluated in tri-modal and bi-modal settings. [*] Training/evaluation with 5 stages at $\pm(20°/150\,\text{cm})$, [$^{\triangle}$] with 2 stages at $\pm(10°/100\,\text{cm})$. T is the average inference time, P is the total number of model parameters, and M is the GPU memory used for inference.

Network	LiDAR-RGB e_t [cm] / e_r [°]	LiDAR-Event e_t [cm] / e_r [°]	T [s]	P [$\times 10^9$]	M [GiB]
		KITTI [6]			
Bi-modal*	2.08 / **0.11**	1.90 / **0.09**	0.51	1.9	14.6
Tri-modal*	**1.80 / 0.11**	**1.82** / 0.12	**0.33**	**1.7**	**11.1**
		DSEC [5]			
Bi-modal$^{\triangle}$	**2.55 / 0.28**	3.43 / **0.18**	0.22	0.8	9.7
Tri-modal$^{\triangle}$	2.62 / 0.30	**2.05** / 0.25	**0.14**	**0.7**	**7.1**
Bi-modal*	**1.40 / 0.12**	1.31 / 0.09	0.44	1.9	14.6
Tri-modal*	2.51 / 0.14	**1.18 / 0.07**	**0.31**	**1.7**	**11.1**

4.6. Ablation Study

We perform ablation studies on the DSEC dataset to analyze the contributions of key architectural components and input processing strategies in LiREC-Net. We report both transla-

Table 4. Ablation showing the effect of LiDAR's point and depth feature fusion on LiDAR-RGB and LiDAR-Event calibration, conducted on the DSEC dataset.

Point Features	Depth Features	LiDAR-RGB e_t [cm] / e_r [°]	LiDAR-Event e_t [cm] / e_r [°]
✓		14.43 / 0.70	14.05 / 0.64
	✓	2.97 / 0.70	2.16 / 0.60
✓	✓	**2.51 / 0.14**	**1.18 / 0.07**

tion and rotation errors for LiDAR-RGB and LiDAR-Event calibration.

We evaluate the impact of LiDAR feature fusion, as shown in Tab. 4. When either the point or depth feature stream is removed, performance drops sharply. Using only point features increases the LiDAR-RGB translation error from 2.51cm to 14.43cm, and the rotation error from 0.14° to 0.70°. Similarly, using only depth features yields 2.97cm / 0.70° for LiDAR-RGB, showing that both geometric and depth cues are necessary for reliable calibration. LiDAR-Event exhibits a similar trend, confirming that the fusion of point and depth features provides complementary information for accurate alignment.

The effect of the scaled projections (SDP and SFP) in LiREC-Net is analyzed in the studies presented in Tab. 5. Removing either the scaled depth projection (SDP) or the scaled feature projection (SFP) degrades performance. Without SDP, the LiDAR-Event error rises to 3.72cm / 0.16°, while removing SFP leads to 1.45cm / 0.07° for LiDAR-Event. Eliminating both projections results in a significant decline (3.35cm / 0.30° for LiDAR-Event), confirming their complementary roles in estimating spatial correspondences. A similar performance drop is observed with LiDAR-RGB when scaled projections are removed.

Finally, different backbone encoders for RGB, event, and the depth-based LiDAR features are compared, as shown in Tab. 6. Replacing the transformer-based MViTV2 [15] with a ResNet [8] degrades performance to 3.04cm / 0.19° for LiDAR-RGB and 2.45cm / 0.16° for LiDAR-Event, underscoring the benefit of global feature modeling for cross-modal alignment. Overall, the full LiREC-Net configuration with fused LiDAR features, scaled projections, and the MViTV2 backbone achieves the best performance of 2.51cm / 0.14° for LiDAR-RGB and 1.18cm / 0.07° for LiDAR-Event.

5. Limitations and Future Work

LiREC-Net currently assumes that the RGB and event cameras are pre-calibrated, *i.e.*, the relative pose $\mathbf{T}^{\mathrm{Ev}\rightarrow\mathrm{RGB}}$ is known. Though our dual perturbation strategy nullifies this condition w.r.t. the LiDAR coordinate system, we found it critical that the camera features share a common coordinate system, when using a shared LiDAR representation. Furthermore, the presented design focuses on three specific sensors (LiDAR, RGB, and Event). As future work, we aim to remove the pre-calibration assumption by jointly estimating $\mathbf{T}^{\mathrm{Ev}\rightarrow\mathrm{RGB}}$ within the pipeline, and to generalize the framework to other sensor combinations beyond the three considered here, potentially incorporating additional modalities, *e.g.*, thermal, radar, or depth.

Table 5. Ablation showing the effect of scaled projections (SDP and SFP) on LiDAR-RGB and LiDAR-Event calibration, conducted on DSEC dataset.

SDP	SFP	LiDAR-RGB e_t [cm] / e_r [°]	LiDAR-Event e_t [cm] / e_r [°]
		3.37 / 0.34	3.35 / 0.30
✓		2.95 / 0.15	1.45 / **0.07**
	✓	4.18 / 0.18	3.72 / 0.16
✓	✓	**2.51 / 0.14**	**1.18 / 0.07**

Table 6. Ablation showing the effect of MViTV2 and ResNet on LiDAR-RGB and LiDAR-Event calibration, conducted on DSEC dataset.

Encoder	LiDAR-RGB e_t [cm] / e_r [°]	LiDAR-Event e_t [cm] / e_r [°]
ResNet [8]	3.04 / 0.19	2.45 / 0.16
MViTV2 [15]	**2.51 / 0.14**	**1.18 / 0.07**

6. Conclusion

In conclusion, we propose LiREC-Net, a target-free, learning-based, tri-modal framework that jointly calibrates LiDAR-RGB and LiDAR-Event pairs in a single architecture. LiREC-Net fuses point- and depth-based LiDAR features in a shared LiDAR branch, employs modality-specific RGB/event encoders, constructs pair-wise cross-modal cost volumes, and predicts translation and unit-quaternion rotation. It's design and training strategy enable LiREC-Net to predict highly competitive and efficient extrinsic calibrations for tri-modal sensory systems.

Acknowledgment

This work was partially funded by the Federal Ministry of Research, Technology, and Space Germany under the project COPPER (16IW24009).

References

[1] Mathieu Cocheteux, Aaron Low, and Marius Bruehlmeier. UniCal: a Single-Branch Transformer-Based Model for

Camera-to-LiDAR Calibration and Validation. *arXiv preprint arXiv:2304.09715*, 2023. 2

[2] Mathieu Cocheteux, Julien Moreau, and Franck Davoine. PseudoCal: Towards Initialisation-Free Deep Learning-Based Camera-LiDAR Self-Calibration. In *The British Machine Vision Conference (BMVC)*, 2023. 1, 2, 6

[3] Mathieu Cocheteux, Julien Moreau, and Franck Davoine. MULi-Ev: Maintaining Unperturbed LiDAR-Event Calibration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024. 1, 2, 5, 6

[4] Ankit Dhall, Kunal Chelani, Vishnu Radhakrishnan, and K. M. Krishna. LiDAR-Camera Calibration using 3D-3D Point correspondences. *arXiv preprint arXiv:1705.09785*, 2017. 1, 2

[5] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A Stereo Event Camera Dataset for Driving Scenarios. *IEEE Robotics and Automation Letters (RA-L)*, 2021. 2, 5, 7

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5, 7

[7] Andreas Geiger, Frank Moosmann, Ömer Car, and Bernhard Schuster. Automatic Camera and Range Sensor Calibration using a single Shot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012. 1, 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[9] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From Video Frames to Realistic DVS Events. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021. 5

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[11] Ganesh Iyer, R. Karnik Ram, J. Krishna Murthy, and K. Madhava Krishna. CalibNet: Geometrically Supervised Extrinsic Calibration using 3D Spatial Transformer Networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 1, 2, 6

[12] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 6

[13] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[14] Yu-Chen Lee and Kuan-Wen Chen. LCCRAFT: LiDAR and Camera Calibration Using Recurrent All-Pairs Field Transforms Without Precise Initial Guess. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 1, 2, 6

[15] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. MViTv2: Improved Multiscale Vision Transformers for Classification and Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 8

[16] Xudong Lv, Boya Wang, Ziwen Dou, Dong Ye, and Shuo Wang. LCCNet: LiDAR and Camera Self-Calibration using Cost Volume Network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021. 1, 2, 4, 5, 6

[17] Gaurav Pandey, James McBride, Silvio Savarese, and Ryan Eustice. Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2012. 1, 2

[18] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. RegNet: Multimodal Sensor Registration Using Deep Neural Networks. In *IEEE Intelligent Vehicles Symposium (IV)*, 2017. 1, 2, 5, 6

[19] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[20] Kevin Ta, David Bruggemann, Tim Brödermann, Christos Sakaridis, and Luc Van Gool. L2E: Lasers to Events for 6-DoF Extrinsic Calibration of Lidars and Event Cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 2

[21] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[22] Shaoan Wang, Zhanhua Xin, Yaoqing Hu, Dongyue Li, Mingzhu Zhu, and Junzhi Yu. EF-Calib: Spatiotemporal Calibration of Event- and Frame-Based Cameras Using Continuous-Time Trajectories. *IEEE Robotics and Automation Letters (RA-L)*, 2024. 2

[23] Xinshuo Weng and Kris Kitani. Monocular 3D Object Detection with Pseudo-LiDAR Point Cloud. In *IEEE/CVF International Conference on Computer Vision Workshops (CVPRW)*, 2019. 2

[24] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler, Faster, Stronger. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3