Robust Long-Range Perception Against Sensor Misalignment in Autonomous Vehicles

Zi-Xiang Xia, Sudeep Fadadu, Yi Shi, Louis Foucard
Aurora Innovation, Inc.
{sxia, sfadadu, yishi, flouis}@aurora.tech

Abstract

Advances in machine learning algorithms for sensor fusion have significantly improved the detection and prediction of other road users, thereby enhancing safety. However, even a small angular displacement in the sensor's placement can cause significant degradation in output, especially at long range. In this paper, we demonstrate a simple yet generic and efficient multi-task learning approach that not only detects misalignment between different sensor modalities but is also robust against them for long-range perception. Along with the amount of misalignment, our method also predicts calibrated uncertainty, which can be useful for filtering and fusing predicted misalignment values over time. In addition, we show that the predicted misalignment parameters can be used for self-correcting input sensor data, further improving the perception performance under sensor misalignment.

1. Introduction

The rapid evolution of machine learning algorithms [3, 27] has revolutionized the field of autonomous vehicles and Advanced Driver Assistance Systems (ADAS), paving the way for safer and more efficient operation on the road. Deep learning techniques, in particular, have proven invaluable in processing and making sense of the vast amounts of data collected from various sensors, including cameras, LiDAR, and radar [31] [8] [18] [10].

The multimodal perception systems combine the strengths of different sensors, providing a more comprehensive and robust understanding of the vehicle's surroundings. For example, camera data offers high-resolution visual information, while LiDAR excels at accurately mapping the 3D environment, and radar is adept at detecting and tracking moving objects even in challenging weather conditions.

Sensor fusion algorithms and models rely on intrinsic and extrinsic calibration parameters to integrate these different sensors. These parameters define the precise spatial relationships and transformations between the various sensors (such as cameras, LiDAR, and radar) and the vehicle's coordinate frame. These parameters are determined through a process known as calibration. Any deviations from these predetermined extrinsic parameters, caused by factors like vibrations, impacts, or thermal effects, can lead to inconsistencies in the sensor data fusion process. Consequently, this can result in erroneous object detection, localization errors, and ultimately, compromised decision-making capabilities.

For self-driving highway applications, especially heavyweight trucks, autonomy is highly susceptible and requires extra attention due to the longer stopping distance requirement. A deviation of just 5 milliradians translates to an error of 2.25 meters at 450 meter range, resulting in a complete mismatch of the fused image and LiDAR data, and enough lateral error to place a vehicle on the wrong lane.

Monitoring the relative positions of onboard sensors can be crucial for safe operations. If the sensors' positions are found to have deviated from their original locations, the planner system of the autonomous vehicle (AV) can take appropriate actions to mitigate risks to acceptable levels, such as stopping on the shoulder or reducing speed. In addition to monitoring, it is essential that the perception models used for detection and tracking tasks are robust against deviations from predetermined calibration parameters.

In this paper, we propose a system that integrates the misalignment monitoring task with a 3D detection task. We limit our work to only angular displacement in this paper as translational displacement (a few millimeters) has minimal impact on sensor measurements compared to the inherent variability (variance) present in the sensor's output.

To achieve this, we integrate a misalignment prediction head to a base detection network-trunk, and we train the network end to end. This solution leverages joint multi-task learning, offering potential benefits in terms of increased robustness of object detection, accurate misalignment prediction, and efficient computational resource utilization. This approach enables the system to self-correct the alignment errors, enhancing the robustness of long-range detection. Our approach leverages aleatoric (data) uncertainty [16],

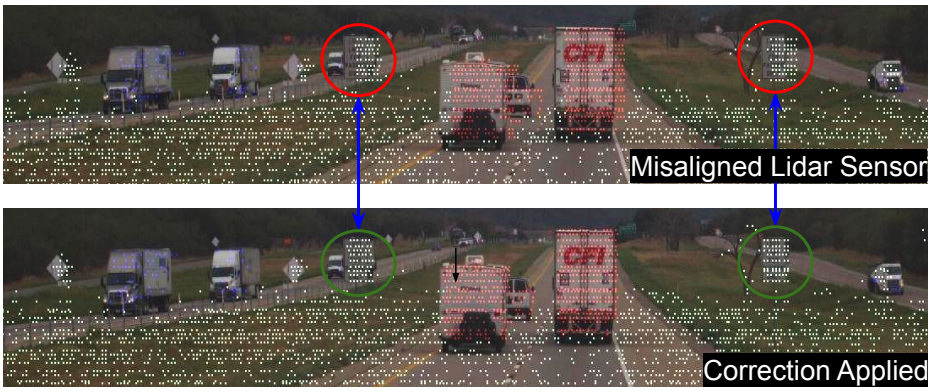

Figure 1. Demonstration of sensor misalignment correction using the proposed approach. The top image shows an input frame with visible misalignment when projecting the LiDAR data (green points) onto the camera image. In the bottom image, after applying the extrinsic parameter correction predicted by our model, the LiDAR points are well-aligned with the camera data, leading to improved 3D object detection accuracy.

which captures the inherent noise and stochasticity in the input sensor data. By explicitly representing this uncertainty in the network's outputs, we can provide well-calibrated confidence estimates for the predicted misalignment values. This enables better-informed decision-making and risk-averse behavior in situations with high uncertainty.

Our contributions are summarized as follows:

- We propose a long-range perception system that is robust against sensor misalignment.
- We propose a learning approach with synthetic data augmentation for the sensor misalignment prediction task where ground-truth data is not readily available.
- We show that learning misalignment as an auxiliary task serves as a valuable enhancement to object detection models.
- Our model predicts well-calibrated uncertainty values for the corresponding misalignment predictions, which can be used further to construct accurate estimate of misalignment over time.
- We show that correcting incoming misaligned data with predicted misalignment helps achieve higher detection performance at longer ranges.

2. Related Work

The extrinsic calibration parameters, a 6-DoF transformation between different sensors, are particularly critical in highway driving due to their operational characteristics. High speed vehicles (specially trucks) require significantly longer stopping distances, necessitating the ability to detect even small objects like a construction cone or a pedestrian at a long range, often up to 450 meters. Performing sensor fusion at these ranges requires a very high degree of precision of each sensor's extrinsics parameters.

2.1. Calibration

Traditionally, LiDAR and camera offline calibration can be categorized into target-based and target-less methods. Target-based calibration, relies on reference objects such as checkerboards [12, 37], ArUco [14], and circular grid [6] to forge a correspondence between LiDAR data and camera images. These methods typically leverage distinctive features like planes and corners to optimize the calibration process, providing high precision within controlled environments. While these methods provide high precision in controlled environments, they are limited by the need for such specific settings. In contrast, target-less methods leverage natural environmental features, such as lines (e.g., lines [1, 28] and edges [21, 35]). Yuan et al. [36] extracted more accurate and reliable LiDAR edges by using voxel cutting and plane fitting techniques. However, maintaining a desired level of calibration precision is challenging due to varying environmental and operational factors such as temperature changes, road vibrations, and equipment wear and tear, all of which can introduce calibration errors on an initially well-calibrated system.

Therefore, online calibration is needed to better handle these conditions on the road. In [22], authors monitored the calibration by computing alignment between image edges and projected LiDAR points. An objective function is proposed to model the information of discontinuity from the image and the depth from LiDAR points. Additionally, inspired by the recent success of deep neural networks, online learning-based approaches are also introduced to calibration. RegNet [32], the first deep convolutional neural

network for LiDAR-camera calibration marks a significant breakthrough in this field. It is trained on a large amount of miscalibration data to accurately estimate extrinsic parameters. Zhu et al. [38] proposed a semantic segmentation method that first extracts semantic features and combines non-monotonic line search and subgradient ascent to estimate optimal calibration parameters. Sun et al. [33] introduced an instance segmentation approach with a cost function that accounts for the matching degree based on the appearance and centroids of key targets in point cloud and image pairs. End-to-end networks, such as CalibNet [15] incorporate geometric supervision to enhance precision, while LCCNet [26] exploits a cost volume layer for feature matching, enabling it to capture correlations between different sensors.

Although online calibration methods have been introduced, it's important to acknowledge that perfect calibration might not be achievable and the perception system must also be robust enough to handle unexpected discrepancies in sensor data. For instance, [15] achieves impressive mean error in yaw orientation estimate. However, it is still high enough that LiDAR points corresponding to a pedestrian at a 400-meter range would be projected laterally onto a different lane. Moreover, these methods are not multi-task methods and require a standalone network alongside a detection network increasing the cost for resource-constrained platforms.

2.2. Robust 3D detection

In the field of 3D object detection, especially when working with LiDAR point clouds, data augmentation is crucial to enhance the robustness and accuracy of the detection models. Scaling, rotation, translation, and flipping of the entire point cloud, are common global augmentation methods [19, 24]. Local augmentations, focusing on individual objects within a point cloud, include techniques like rotation, scaling, and translation of specific ground truth boxes [13]. Moreover, innovative techniques like PointMixup [2], which employs interpolation between point clouds, and InverseAug [23], which focuses on reversing augmented features and understanding the correlations between LiDAR and camera features during fusion, have been introduced. Each method offers a unique contribution to enhancing LiDAR point cloud data when leveraging multi-modality fusion, leading to more accurate and dependable 3D object detection. However, none of these methods explicitly perform perturbation of the extrinsic parameters directly. In our method, we argue that, in addition to the mentioned augmentation and feature interpolation, explicitly perturbing the relative transformation can provide enhanced robustness against sensor misalignment.

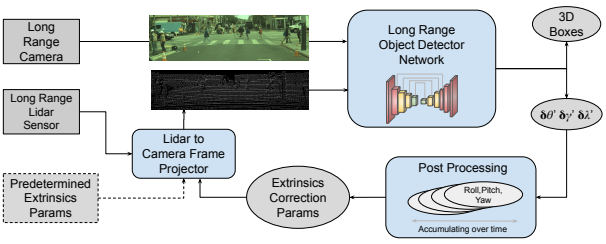

Figure 2. Proposed system architecture for robust long-range perception against sensor misalignment. In the figure, rectangles indicate the system inputs, while circles denote the outputs.

3. Proposed Approach

Our long-range detection system is composed of three phases as described in Figure 2. We first start with transforming the incoming LiDAR point-cloud data from LiDAR's sensor frame of reference into the camera's frame of reference using the pre-determined extrinsic parameters. Next, we run inference with our multi-task detector and misalignment predictor on the data as described in section 3.2. The estimated misalignment parameters, along with predicted uncertainty are fused over a 5-second window. Finally, the fused misalignment values from history are used in the initial pre-processing phase for correcting transformed points in the camera's frame of current reference.

In the following sections, we describe each phase in more detail, including model architecture, loss, training details, and the self-correction process.

3.1. Phase 1: Pre-Processing

In this phase, we prepare a depth image from the LiDAR point-cloud data. Given extrinsic T and intrinsic K, we can generate the depth image by projecting LiDAR points $P_i = [X_i\ Y_i\ Z_i] \in \mathbb{R}^3$ to image pixel coordinate $p_i = [u_i\ v_i] \in \mathbb{R}^2$. The pixel values are set to Z_i where the LiDAR point is projected. For the remaining pixels, the value is set to zero.

3.2. Phase 2: Inference

3.2.1 Misalignment Estimation

In the event of pure rotational displacement of LiDAR sensor, the coordinates of the point-cloud shift accordingly. For rotational displacement of $\Delta r = [\theta^{roll}, \theta^{pitch}, \theta^{yaw}]$, the projected pixel coordinates in depth image I_{depth}, will be shifted to $p'_i = [u'_i\ v'_i] \in \mathbb{R}^2$. compared to the corresponding pixel locations p_i in the RGB image I_{rgb}. Since the relative translation component in T is negligible compared to the long ranges we are interested in, we approximate the problem as one of pure rotation with respect to the camera's

frame of reference. With this approximation, values of p'_i can be obtained from p_i using a Homography matrix H.

$$\begin{aligned} \hat{p}'_i &\approx H \cdot \hat{p}_i \\ &\approx K \cdot \Delta R \cdot K^{-1} \cdot \hat{p}_i \end{aligned} \tag{1}$$

where $\hat{p}_i$ and $\hat{p}'_i$ represent the homogeneous coordinates of p_i and p'_i, ΔR is the rotation matrix representation of relative rotation Δr between I_{depth} and I_{rgb}.

Throughout the research, the intrinsic matrix K is known and fixed. To further simplify the equation, We also assume that the Δr is sufficiently small, and small angle approximation [5] can be applied. With $\tilde{p}_i = K^{-1}p_i = [\tilde{x}_i, \tilde{y}_i, \tilde{z}_i]$ and $\tilde{p}'_i = K^{-1}p'_i = [\tilde{x}'_i, \tilde{y}'_i, \tilde{z}'_i]$, the equation 1 can be rewritten as,

$$\begin{aligned} \begin{bmatrix} \tilde{x}'_i \\ \tilde{y}'_i \\ \tilde{z}'_i \end{bmatrix} &\approx \begin{bmatrix} 1 & -\theta^{yaw} & \theta^{pitch} \\ \theta^{yaw} & 1 & -\theta^{roll} \\ -\theta^{pitch} & \theta^{roll} & 1 \end{bmatrix} \cdot \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \\ \tilde{z}_i \end{bmatrix} \\ &\approx \begin{bmatrix} \tilde{x}_i - \theta^{yaw}\tilde{y}_i + \theta^{pitch}\tilde{z}_i \\ \theta^{yaw}\tilde{x}_i + \tilde{y}_i - \theta^{roll}\tilde{z}_i \\ -\theta^{pitch}\tilde{x}_i + \theta^{roll}\tilde{y}_i + \tilde{z}_i \end{bmatrix} \end{aligned} \tag{2}$$

Given the point location correspondences $\tilde{p}_i$ and $\tilde{p'}_i$, we can set up a linear system $Ax = 0$, where $x = [\theta^{roll}\ \theta^{pitch}\ \theta^{yaw}\ 1]^T$.

$$\begin{bmatrix} 0 & \tilde{z}_1 & -\tilde{y}_1 & \tilde{x}_1 - \tilde{x}'_1 \\ -\tilde{z}_1 & 0 & \tilde{x}_1 & \tilde{y}_1 - \tilde{y}'_1 \\ \tilde{y}_1 & -\tilde{x}_1 & 0 & \tilde{z}_1 - \tilde{z}'_1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \tilde{z}_n & -\tilde{y}_n & \tilde{x}_n - \tilde{x}'_n \\ -\tilde{z}_n & 0 & \tilde{x}_n & \tilde{y}_n - \tilde{y}'_n \\ \tilde{y}_n & -\tilde{x}_n & 0 & \tilde{z}_n - \tilde{z}'_n \end{bmatrix} \begin{bmatrix} \theta^{roll} \\ \theta^{pitch} \\ \theta^{yaw} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

However, obtaining correspondences between I_{rgb} and I_{depth} is a non-trivial task. To address this, we employ multi-task learning to directly regress Δr in a similar fashion as demonstrated in [30] for fundamental matrix estimation without known correspondences. For ground-truth Δr, we perturb the LiDAR points in the camera image frame of reference and train the network to estimate the amount of perturbation. Also note that throughout this work, we assume that the intrinsic matrix does not change significantly between the training and testing sets which makes the matrix A dependent only on the input RGB and depth images.

3.2.2 Network Architecture

To demonstrate the effectiveness of our method, we use CenterNet [7] as our base model. CenterNet consists of a convolutional backbone network for feature extraction and decoding heads to output heatmaps and offsets. In this work, we adopt a similar structure, employing VoVNetV2 [20] as the CNN backbone. Initially, the image is processed through a stem network consisting of two fully convolutional layers with 32 and 64 dimensions, and kernel sizes of 7 × 7 and 3 × 3, respectively. The first layer utilizes a stride of 2, reducing the feature resolution to half the original. The LiDAR depth image is then downsampled to half resolution and concatenated with the image feature map. This combined feature map is fed into the VoVNetV2 feature extractor, which operates through eight stages. The first four stages perform a 2× downsampling, while the subsequent stages upsample the feature map back to half resolution. At each up-sampling stage, the depth image is resized and concatenated with the feature map before being fed into the next stage. Finally, the feature maps are passed through a 1 × 1 convolution to decode the outputs. For object category (user-defined channels), 2D center (2 channels), 2D size (2 channels), 3D centroid (3 channels), 3D size (3 channels), range (1 channel), and orientation (1 channel), each output heatmap is of size (H/2 × W/2) × individual channels. Additionally, the head produces the LiDAR miscalibration output, which consists of three scalar values representing pitch, yaw, and roll in degrees.

3.2.3 Loss Functions

Below, we iterate over all the loss functions used to train the multi-task network mentioned in the previous section. We apply different weights $W_{\{c,o,s,\hat{o},\hat{s},d,\phi,\theta\}}$ on each loss for multi-task learning.

First, we use focal loss [25] to supervise the category output:

$$\mathcal{L}_{\text{class}} = -\frac{W_c}{N}\sum_{i}^{N}\alpha_i(1-p_i)^{\gamma}\log(p_i) \tag{4}$$

where N denotes the number of pixels, p_i represents the predicted probability for the ground-truth class at pixel i, and (α, γ) are focal loss weight parameters.

In addition to classification loss, for reliable 2D detection location and extent prediction; we supervise the offsets (o_i^x, o_i^y) of the heatmap pixel, the width w_j^{2D} and the height h_j^{2D} of each object bounding box j. To handle uncertainties, the network predicts both the mean and the diversity of a Laplacian distribution. The loss function is applied to minimize the negative log-likelihood of the Laplacian distribution [4]. The diversity parameter b is used as the uncertainty value [29].

Here * denotes the model's predictions, and $b_{\{o_i^x, o_i^y, w_i^{\text{2D}}, h_i^{\text{2D}}\}}$ is the diversity parameter for each distribution [4].

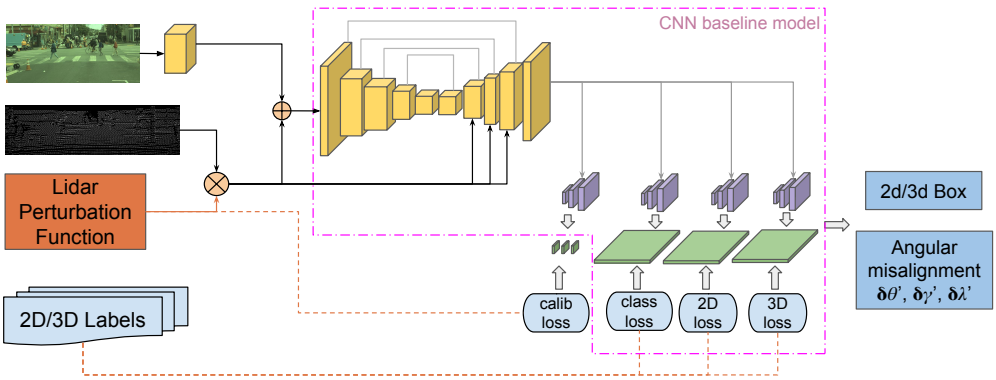

Figure 3. Our proposed multi-task network architecture and data augmentation approach. The network takes in RGB image and LiDAR points projected onto a virtual image. During the training, projected LiDAR points are perturbed in a controlled fashion, and the network is tasked to predict the parameters of that perturbation along with detecting and classifying 3D objects in the scene. This synthetic data augmentation and training technique ensures the model is robust to small perturbations in real-world scenarios.

$$\mathcal{L}_{\text{2D}} = \frac{1}{N}\sum_{i=1}^{N}\Bigg(W_o\Big(\frac{\|o_i^x - o_i^{x*}\|_1}{b_{o_i^x}} + \frac{\|o_i^y - o_i^{y*}\|_1}{b_{o_i^y}}\Big) + W_s\Big(\frac{\|w_i^{\text{2D}} - w_i^{\text{2D}*}\|_1}{b_{w_i^{\text{2D}}}} + \frac{\|h_i^{\text{2D}} - h_i^{\text{2D}*}\|_1}{b_{h_i^{\text{2D}}}}\Big) + W_o \log(b_{o_i^x} b_{o_i^y}) + W_s \log(b_{w_i^{\text{2D}}} b_{h_i^{\text{2D}}})\Bigg) \tag{5}$$

Similarly, to accurately estimate the position of the detected object in 3D, the losses $\mathcal{L}_{\text{3D}}$ for the 3D position and extent parameters are computed in an analogous way. We compute the offsets $(\hat{o}_i^x, \hat{o}_i^y)$ of the heatmap pixel by projecting the 3D box centroid onto 2D. We use L1 loss to supervise the 3D Euclidean distance d_i and the orientation ϕ_i.

$$\mathcal{L}_{\text{3D}} = \frac{1}{N}\sum_{i=1}^{N}\Bigg(W_{\hat{o}}\Big(\frac{\|\hat{o}_i^x - \hat{o}_i^{x*}\|_1}{b_{\hat{o}_i^x}} + \frac{\|\hat{o}_i^y - \hat{o}_i^{y*}\|_1}{b_{\hat{o}_i^y}}\Big) + W_{\hat{s}}\Big(\frac{\|w_i^{\text{3D}} - w_i^{\text{3D}*}\|_1}{b_{w_i^{\text{3D}}}} + \frac{\|l_i^{\text{3D}} - l_i^{\text{3D}*}\|_1}{b_{l_i^{\text{3D}}}} + \frac{\|h_i^{\text{3D}} - h_i^{\text{3D}*}\|_1}{b_{h_i^{\text{3D}}}}\Big) + W_{\hat{o}} \log(b_{\hat{o}_i^x} b_{\hat{o}_i^y}) + W_{\hat{s}} \log(b_{w_i^{\text{3D}}} b_{h_i^{\text{3D}}} b_{l_i^{\text{3D}}}) + W_d\|d_i - d_i^*\|_1 + W_\phi\|\phi_i - \phi_i^*\|_1\Bigg) \tag{6}$$

The miscalibration loss is supervised in pitch, yaw, and roll axes as shown in equation 7:

$$\mathcal{L}_{\text{miscal}} = W_\theta\Bigg(\frac{\|\theta^{\text{pitch}} - \theta^{\text{pitch}*}\|_1}{b_{\theta^{\text{pitch}}}} + \frac{\|\theta^{\text{yaw}} - \theta^{\text{yaw}*}\|_1}{b_{\theta^{\text{yaw}}}} + \frac{\|\theta^{\text{roll}} - \theta^{\text{roll}*}\|_1}{b_{\theta^{\text{roll}}}} + \log(b_{\theta^{\text{pitch}}} b_{\theta^{\text{yaw}}} b_{\theta^{\text{roll}}})\Bigg) \tag{7}$$

Finally, the total loss is formed as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{class}} + \mathcal{L}_{\text{2D}} + \mathcal{L}_{\text{3D}} + \mathcal{L}_{\text{miscal}} \tag{8}$$

We choose multi-task weights $W_o = 1.0, W_c = 2.0, W_s = 0.1, W_{\hat{o}} = 0.25, W_{\hat{s}} = 1.0, W_d = 1.5, W_\phi = 0.1$, and $W_\theta = 0.4$ for the experiments.

3.2.4 Training Details

To ensure that our network is robust against sensor misalignment and also predicts LiDAR misalignment accurately, we apply LiDAR perturbation (fault injection) during training. We use Gaussian distribution with a standard deviation of 0.5 degrees to perturb the rotation of LiDAR points within the camera frame along each axis of rotations. These perturbations affect the pitch, yaw, and roll axes up to a maximum of 1.0 degree.

Our method is trained on inputs comprising a single image and 100ms of LiDAR data centered on the image timestamp. Training is performed for 450k iterations using an Adam optimizer [17] with an initial learning rate of 8e-4, decaying by 0.9 every 4000 iterations. We leverage NVIDIA A10 GPUs on AWS g5.4xlarge instances for training.

3.3. Phase 3: Post-Processing

For the final phase of our system, we process the output from the network to obtain 3D detection and accurate misalignment estimation.

3.3.1 3D Detections

To leverage the outputs of our network, which include per-pixel classification, 2D object detection, and 3D object detection heads, we employ a 3D version non-maximum suppression (NMS) strategy similar to CenterNet [7].

3.3.2 Misalignment Fusion

We leverage the predicted uncertainty (diversity parameter $b_{\theta^{\text{pitch}}}$, $b_{\theta^{\text{yaw}}}$, and $b_{\theta^{\text{roll}}}$) values from network to fuse the misalignment estimates over a 5-second window. First, we filter out all the predictions with an uncertainty value higher than 0.3 degree. Then, we perform a weighted average over the remaining measurements.

3.3.3 Self Correction

At module level, we use the fused prediction and apply a correction to incoming LiDAR data to align with camera image input by updating the extrinsic parameters.

$$T_{corrected} = T \cdot [R_{estimate}|0] \tag{9}$$

where $R_{estimate}$ is the rotational matrix corresponding to the fused $[\theta^{roll}, \theta^{pitch}, \theta^{yaw}]$ predictions and translation component is set to 0. During the future iteration $T_{corrected}$ is used in the phase-1 for 3D point projection onto virtual image I_{depth}. In section 4.2, we demonstrate that feeding the model corrected data improves the model performance on critical object detection tasks.

4. Verification

We employ a fault injection methodology that simulates misalignment conditions by perturbing the extrinsic parameters on a carefully curated test dataset. Specifically, we introduce controlled misalignment by altering the extrinsic parameters of the LiDAR sensor, which defines its spatial orientation and position relative to the vehicle's coordinate frame. These faults are injected by perturbing the roll, pitch, and yaw angles within a range of -1.0 to 1.0 degrees of rotation. This range is selected to emulate realistic scenarios where minor misalignments can occur due to factors such as vibrations, impacts, or thermal effects. In the experiment section, we show such minor misalignment is enough to have a severe impact on long range perception (45% regression in 300-400 meters from Table 3).

By systematically injecting these perturbations across the test dataset, we can comprehensively evaluate the performance of our proposed method in detecting and quantifying the misalignment, as well as assess the effectiveness of the mitigation strategies, such as self-correction of extrinsic parameters.

4.1. Experimental Setting

4.1.1 Dataset

To validate our proposed approach, we leverage our diverse internal dataset. Our focus in this study is mainly the degradation in long-range detections due to small misalignments. This limits us from using publicly available datasets such as Waymo Dataset which provides 3D labels only until 85 meters range [34] and KITTI [11] around 70 meters. To provide better reproducibility and to show applicability of this approach for short range perception, we also conduct experiments on Waymo dataset and compare the results against available approaches. The limitations will be further discussed in the following sections.

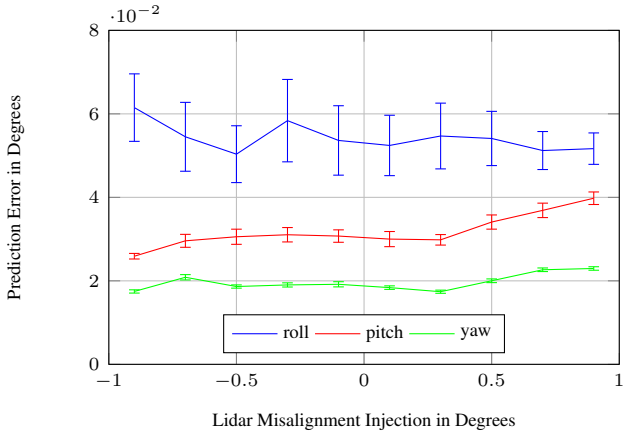

Figure 4. Mean and variance observed in misalignment estimation error at different levels of perturbations for each rotational dimension.

Our internal proprietary dataset provides 3D bounding box labels up to 500m range using Frequency Modulated Continuous Wave (FMCW) LiDAR system. It comprises 43,500 five-second snippets for training and 1,000 for validation, featuring synchronized 8MP camera images with a 30° field of view, LiDAR data from FMCW LiDAR system. More importantly, this dataset enables us to verify our method's effectiveness on distant objects (upto 500 meter range) under diverse conditions. From the dataset, we extract 5-second long continuous snippets of sensor data stream. To simulate realistic misalignment scenarios, we employ purpose built fault injector tool to perturb the incoming LiDAR point data within each snippet, introducing roll, pitch and yaw perturbations uniformly sampled in the range of [-1 degree to 1 degree] with an interval of 0.1 degree with total 1000 snippets of data.

4.1.2 Evaluation Metrics

Here we describe different metrics used to demonstrate the effectiveness of our approach. For the internal long-range dataset, we apply a 1.0 degree LiDAR perturbation, as previously described. For the Waymo dataset, a 5.0 degree perturbation is applied to create more pronounced misalignment in short-range scenarios.

Misalignment Detection Accuracy: The misalignment detection accuracy metric (see MDA columns in Table 1) evaluates our method's ability to reliably identify sensor

On Internal Proprietary Long Range Dataset with $\delta \in [-1°, 1°]$					
Fusion Approach	**Misalig. Acc.**		**Mean Est. Error**		
	Prec. ↑	**Rec.** ↑	**Roll** ↓	**Pitch** ↓	**Yaw** ↓
Per Frame	0.9813	0.9637	0.0700° ±0.017	0.0313° ±0.008	0.0241° ±0.006
Snippet (w/o uncertainty)	0.9821	0.9644	0.0620° ±0.015	0.0298° ±0.005	0.0184° ±0.006
Snippet (w uncertainty)	0.9861	0.9650	0.0450° ±0.008	0.0290° ±0.001	0.0141° ±0.0007

Table 1. Performance metrics with CenterNet baseline [7] per frame, per snippet with fused predictions without considering uncertainty, and per snippet with fused predictions considering uncertainty (filtering out estimates with high predicted uncertainty and performing weighted average over remaining frames).

Method	**Pitch°**	**Yaw°**	**Roll°**
Yuan [36]	0.411	0.325	0.423
Zhu [38]	0.421	0.301	0.305
Sun [33]	**0.202**	0.211	**0.171**
Ours	0.248	**0.068**	0.309

Table 2. Comparison of different methods for pitch, yaw, and roll error on Waymo dataset [34].

alignment degradation while minimizing false alarms. We evaluate results as **positive** when our method predicts misalignment of > 0.1 degree and **negative** when our method predicts misalignment of < 0.1 degree. **True-Positive** is considered, when both predicted and injected misalignment are > 0.1 degree and **True-Negative** is considered when both predicted and injected misalignment are < 0.1 degrees. Using this, we report recall and precision across all snippets under varied conditions to comprehensively assess misalignment detection performance.

Misalignment Estimation Error Analysis: We evaluate the misalignment estimation error at both the per-frame and per-snippet levels. For per-frame evaluation, we directly compare the predicted misalignment values against the injected ground truth misalignment. For per-snippet evaluation, we compute the mean of all frame-level misalignment predictions within a snippet. Additionally, we leverage the predicted uncertainty estimates to fuse the outputs by ignoring predictions with high uncertainty (sigma > 0.3 degrees) and computing a weighted average over the remaining predictions as described in section 3.3.

Misalignment Estimation Error Benchmark: To better evaluate the effectiveness of our approach, we compare it with other target-less learning methods on the Waymo dataset [34]. We leverage the experiment settings from Sun et al. [33], where perturbations are applied differently at each group, and the mean average error is calculated as the final result. However, we compute the average over 30,000 frames, as opposed to the 60 frames in the original setting [33], since we observed result fluctuations across different groups. The comparison is presented in Table 2.

Perception Performance Metrics: In addition to CenterNet, we include SpotNet [9] in this experiment to demonstrate the effectiveness of the proposed method as an auxiliary task. SpotNet is a long range, image-centric, and LiDAR-anchored approach that jointly learns 2D and 3D detection tasks. We measure the 3D detection performance of our multi-task model against a baseline and ensure that our module-level mitigation method strictly increases perception performance across all conditions (see Table 3). We employ the max F1 score metric (IoU=0.1) to evaluate the Bird's Eye View (BEV) object detection performance for actors at different ranges under various fault injection scenarios. As a baseline, we show the result from the model that is not trained with any perturbation. We compare the results of our proposed network architecture with and without applying the correction. This comprehensive evaluation helps us demonstrate the tangible benefits of our mitigation strategy in improving perception robustness and maintaining reliable operation in the presence of sensor misalignment faults.

Additionally, for the Waymo dataset, we include only the results from SpotNet due to a label association issue. In Waymo, the 2D and 3D labels are not linked, preventing us from directly associating 2D labels with their corresponding 3D labels. As a result, we are unable to train CenterNet on 3D detection, since our method relies on 2D labels to encode target image pixels and then trains the 3D attributes from the associated 3D labels. Therefore, we compare only the 2D detection task on the Waymo dataset (see Table 4).

4.2. Results

This section evaluates the performance of our proposed approach and its impact on enhancing safety across various driving scenarios. For system-level failure mitigation, we focus on the Misalignment Detection Accuracy (MDA) metrics, specifically precision and recall. As shown in Table 1, the results demonstrate the high precision and recall achieved by our method in detecting sensor misalignment across different conditions. Additionally, incorporating uncertainty estimates in the Per Snippet evaluation improves misalignment estimation accuracy, as evidenced by the lower Mean Estimated Error values.

Table 2 presents results that are comparable to the benchmark in pitch and roll, while outperforming in yaw. By combining these table results with the misalignment estimates shown in the plot (Figure 4) of the injection LiDAR misalignment, we see that the model demonstrates promising performance in estimating yaw misalignment but struggles with roll misalignment. This behavior could be due to the differences in vertical and horizontal resolution for the LiDAR. LiDAR has a slightly higher horizontal reso-

BEV Max-F1 @ 0.1 IoU for Vehicles with CenterNet [7] ↑						
Range [m]	**Baseline**		**Proposed (No Correction Applied)**		**Proposed (Predicted Correction Applied)**	
	calib	**miscal**	**calib**	**miscal**	**calib**	**miscal**
200-300	0.177	0.104	**0.250 (+41.2%)**	0.249 (+39.4%)	**0.250 (+41.2%)**	**0.251 (+41.3%)**
300-400	0.111	0.061	0.168 (+51.3%)	0.169 (+77.0%)	**0.170 (+53.13%)**	**0.170 (+78.6%)**
400-500	0.082	0.075	0.156 (+90.2%)	0.156 (+90.2%)	**0.161 (+96.3%)**	**0.159 (+112%)**
***BEV Max-F1 @ 0.1 IoU for Vehicles with SpotNet [9] ↑**						
Range [m]	**Baseline**		**Proposed (No Correction Applied)**		**Proposed (Predicted Correction Applied)**	
	calib	**miscal**	**calib**	**miscal**	**calib**	**miscal**
200-300	**0.804**	0.360	0.794 (-1.2%)	0.789 (+119%)	0.789 (-1.8%)	**0.792 (+124.3%)**
300-400	**0.734**	0.250	0.726 (-1.0%)	0.718 (+187%)	0.726 (-1.0%)	**0.721 (+188%)**
400-500	**0.640**	0.220	0.620 (-3.12%)	0.601 (+173%)	0.626 (-3.06%)	**0.626 (+184%)**

Table 3. Comparison of 3D vehicle detection performance. The "Proposed (No Correction Applied)" column represents the performance of our multi-task model when evaluated on miscalibrated (**miscal**) and calibrated (**calib**) data without applying any correction. The "Proposed (Predicted Correction Applied)" column shows the improved performance achieved by applying the extrinsic parameter correction using the misalignment values predicted by our model. For readers, we highlight the highest score in each row with bold characters for calibrated and miscalibrated test data category individually. **The dataset count slightly differs.*

lution, which makes estimating yaw miscalibration easier compared to Pitch. The roll noise is only visible at the very edge of the image, which typically consists of featureless areas such as the sky or the ground surface closer to the camera (see Figure 1), making it more difficult for the model to observe feature displacement.

Table 3 compares the performance of our multi-task model and a baseline model for vehicle detection on calibrated and miscalibrated data. We observe that the multi-task model significantly improves detection performance compared to the baseline when evaluated on the same miscalibrated data. It is important to note the significant performance drop in SpotNet [9] under miscalibrated scenarios. This is because SpotNet is a LiDAR-anchored approach, making it highly sensitive to miscalibration. When there is misalignment between the projected sensor points and the image, the model mismatches the sensor data with the image background features, leading to decreased accuracy. Furthermore, when applying correction (see qualitative example in Figure 1) using the predicted misalignment values, the performance improves for ranges beyond 200 meters. Notably, the model already performs adequately without the calibration correction, as it is trained on miscalibrated data, making it inherently robust against misalignment errors and thus reducing the impact of the correction mechanism. Table 4 also shows that the proposed method improves short range 2D detection on the Waymo dataset

We also note that, even on the well-calibrated test set, the performance of the proposed model increases compared to the baseline; which was unexpected earlier. Upon closer investigation, we notice that in many instances our manually curated dataset has misalignment that went unnoticed. In a few cases, the alignment was accurate, and model improved in ultra long-range detections (400 meters and beyond), which further shows that LiDAR perturbation could be a powerful tool as augmentation for training.

On Short Range Waymo Dataset [34] with SpotNet [9] with $\delta \in [-5^\circ, 5^\circ]$		
Approach	**Eval Data**	**Max-F1 for 2D Detection**
Baseline	Calibrated	0.693
	MisCalibrated	0.640
Proposed w/o input correction	Calibrated	**0.712**
	MisCalibrated	**0.698**

Table 4. 2D detection performance metrics shows improvement on short-range perception with our proposed augmentation learning approach.(note: For short-range waymo dataset, we inject $\delta \in [-5, 5]$ degree of miscalibration)

5. Conclusion

In this research, we addressed the problem of making perception systems in autonomous vehicles robust against sensor misalignment. We proposed using a multi-task network trained with synthetic data augmentation to predict the degree of misalignment between LiDAR and camera sensors, alongside the detection task. We fuse the predictions using predicted data uncertainty and use the final output for self-correcting the input data. Finally, we evaluated the method on a large-scale, real-world dataset, perturbed with a purpose-built fault injection method, to demonstrate the effectiveness of our proposal.

Acknowledgments

We express our special thanks to Chi-Kuei Liu and Thuyen Ngo[1] for their invaluable contributions.

[1]For their help in developing the model and preparing the dataset.

References

[1] Ziqi Chai, Yuxin Sun, and Zhenhua Xiong. A novel method for lidar camera calibration by plane fitting. In *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 286–291, 2018. 2

[2] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees G. M. Snoek. Pointmixup: Augmentation for point clouds. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 330–345, Cham, 2020. Springer International Publishing. 3

[3] Pranav Singh Chib and Pravendra Singh. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *IEEE Transactions on Intelligent Vehicles*, 2023. 1

[4] Wikipedia contributors. Log-laplace distribution, 4 2024. 4

[5] Wikipedia contributors. Small-angle approximation, 4 2024. 4

[6] Joris Domhof, Julian F.P. Kooij, and Dariu M. Gavrila. An extrinsic calibration tool for radar, camera and lidar. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8107–8113, 2019. 2

[7] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577, 2019. 4, 5, 7, 8

[8] Sudeep Fadadu, Shreyash Pandey, Darshan Hegde, Yi Shi, Fang-Chieh Chou, Nemanja Djuric, and Carlos Vallespi-Gonzalez. Multi-view fusion of sensor data for improved perception and prediction in autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2349–2357, January 2022. 1

[9] Louis Foucard, Samar Khanna, Yi Shi, Chi-Kuei Liu, Quinn Z Shen, Thuyen Ngo, and Zi-Xiang Xia. Spotnet: An image centric, lidar anchored approach to long range perception. *arXiv preprint arXiv:2405.15843*, 2024. 7, 8

[10] Shivam Gautam, Gregory P. Meyer, Carlos Vallespi-Gonzalez, and Brian C. Becker. Sdvtracker: Real-time multi-sensor association and tracking for self-driving vehicles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3012–3021, October 2021. 1

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 6

[12] Andreas Geiger, Frank Moosmann, Ömer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *2012 IEEE International Conference on Robotics and Automation*, pages 3936–3943, 2012. 2

[13] Jordan S.K. Hu and Steven L. Waslander. Pattern-aware data augmentation for lidar 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2703–2710, 2021. 3

[14] Jiunn-Kai Huang and Jessy W. Grizzle. Improvements to target-based 3d lidar to camera calibration. *IEEE Access*, 8:134101–134110, 2020. 2

[15] Ganesh Iyer, R. Karnik Ram, J. Krishna Murthy, and K. Madhava Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1110–1117, 2018. 3

[16] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 1

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Ankit Laddha, Shivam Gautam, Stefan Palombo, Shreyash Pandey, and Carlos Vallespi-Gonzalez. Mvfusenet: Improving end-to-end object detection and motion forecasting through multi-view fusion of lidar data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2865–2874, 2021. 1

[19] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2019. 3

[20] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 752–760, 2019. 4

[21] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*, 2013. 2

[22] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. 06 2013. 2

[23] Y. Li, A. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le, A. Yuille, and M. Tan. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17161–17170, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society. 3

[24] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 663–678, Cham, 2018. Springer International Publishing. 3

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. 4

[26] Xudong Lv, Boya Wang, Ziwen Dou, Dong Ye, and Shuo Wang. Lccnet: Lidar and camera self-calibration using cost volume network. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2888–2895, 2021. 3

[27] Yifang Ma, Zhenyu Wang, Hong Yang, and Lin Yang. Artificial intelligence applications in the development of autonomous vehicles: A survey. *IEEE/CAA Journal of Automatica Sinica*, 7(2):315–329, 2020. 1

[28] Peyman Moghadam, Michael Bosse, and Robert Zlot. Line-based extrinsic calibration of range and image sensors. In *2013 IEEE International Conference on Robotics and Automation*, pages 3685–3691, 2013. 2

[29] Deebul S Nair, Nico Hochgeschwender, and Miguel A Olivares-Mendez. Maximum likelihood uncertainty estimation: Robustness to outliers. *arXiv preprint arXiv:2202.03870*, 2022. 4

[30] Omid Poursaeed, Guandao Yang, Aditya Prakash, Qiuren Fang, Hanqing Jiang, Bharath Hariharan, and Serge Belongie. Deep fundamental matrix estimation without correspondences. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 4

[31] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7077–7087, 2021. 1

[32] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1803–1810, 2017. 2

[33] Chao Sun, Zhijie Wei, Wenyi Huang, Qianfei Liu, and Bo Wang. Automatic targetless calibration for lidar and camera based on instance segmentation. *IEEE Robotics and Automation Letters*, 8(2):981–988, 2023. 3, 7

[34] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 6, 7, 8

[35] Chongjian Yuan, Xiyuan Liu, Xiaoping Hong, and Fu Zhang. Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robotics and Automation Letters*, 6(4):7517–7524, 2021. 2

[36] Chongjian Yuan, Xiyuan Liu, Xiaoping Hong, and Fu Zhang. Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robotics and Automation Letters*, 6(4):7517–7524, 2021. 2, 7

[37] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 3:2301–2306 vol.3, 2004. 2

[38] Yufeng Zhu, Chenghui Li, and Yubo Zhang. Online camera-lidar calibration with sensor semantic information. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4970–4976, 2020. 3, 7