Cross-dataset Sensor Alignment: Making Visual 3D Object Detector Generalize

Anonymous Author(s)
Affiliation
Address
email

Abstract: While camera-based 3D object detection has been rapidly evolving, these models are prone to overfit to different kinds of spurious artifact, such as sensor suite and vehicle sizes. In fact, most datasets for autonomous driving are collected using a single sensor configuration, and the data collection is limited to a few specific cities and similar weather conditions. Therefore, these camera-based models do not generalize to scenarios in the wild. To address that, this paper introduces a general approach to unify all existing datasets to evaluate the generalization capability of camera-based 3D object detectors, including adapting detectors from one dataset to another and training detectors with multiple datasets. We observe that simply aggregating datasets and test a 3D object detector yield drastic performance drop, even with more data available during training. We conducted extensive experiments to investigate the key issues that lead to this gap. Our results reveal that the major challenge comes from the deviations in sensor configuration, which can not be easily learned by deep models even with diverse data. To close the gap, we propose a simple yet effective method to align different sensor settings—a combination of camera intrinsic correction, camera extrinsic correction, and ego frame alignment. Our method achieves strong cross-dataset performance boosting. Compared with single dataset baselines, we achieve 42.4 mAP improvement on KITTI, 24.1 mAP improvement on Lyft, 17.9 mAP improvement on KITTI-360, 17.1 mAP improvement on nuScenes, 7.7 mAP improvement on Argoverse2 and 3.6 mAP improvement on Waymo. We hope this comprehensive study can facilitate research on generalizable 3D object detection and adjacent directions.

Keywords: 3D object detection, Model Generalization, Autonomous Driving

1 Introduction

Autonomous vehicles require precise 3D detection and localization of traffic participants, such as cars, pedestrians, and bicycles, to ensure safe driving. Consequently, 3D object detection has garnered significant attention, leading to improved accuracy on various benchmarks [1, 2, 3]. However, existing methods [4, 5, 6, 7, 8, 9, 10] often train and test on the same benchmark, overlooking the impact of data diversity and assuming that the training and testing data have the same distribution. This assumption may not hold true in realistic scenarios, for instance, when a detection algorithm is deployed across dozens of different vehicles (*e.g.* Volkswagen Golf or Ford F-series). Thus, it is questionable whether they can effectively generalize to and learn from a variety of datasets.

To answer this question, we first conduct a simple experiment that trains a model on one dataset and tests it on another dataset. The results revealed an unexpected drastic performance degradation: a detector trained on the Argoverse2 [2] dataset experiences a 70.4% performance drop when tested on the Waymo [3] dataset compared to the one trained on Waymo. Then, we add the nuScenes [11] dataset to our experiment to improve training data volume. However, the model with additional data

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

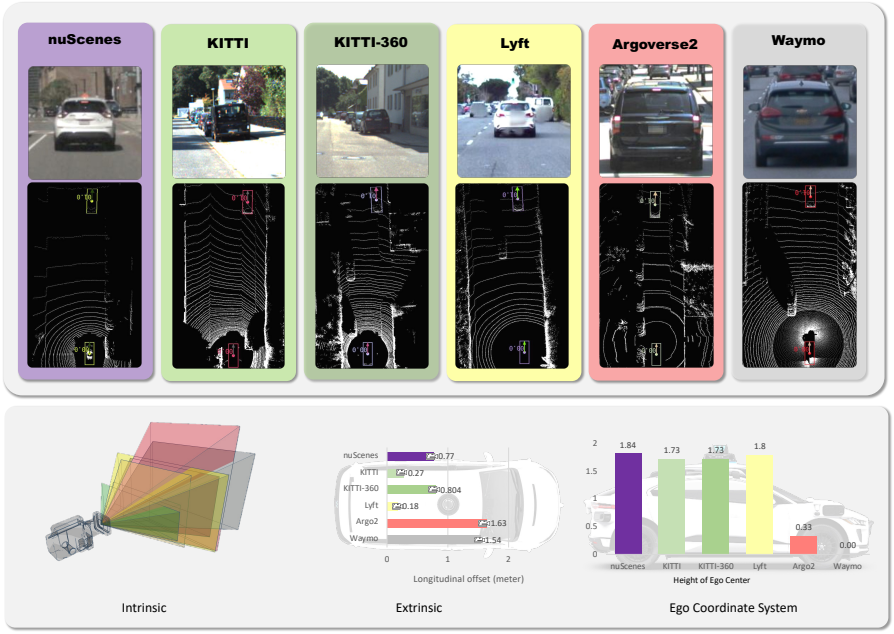


Figure 1: Top: **Similar cars in different datasets.** We provide examples of cars that exhibit similar sizes and distances from the ego vehicle in six datasets. The top row showcases image patches, cropped from original images with dimensions of 160x160 pixels. The bottom row displays the corresponding LiDAR point clouds, consistently scaled across all datasets. However, due to variations in camera configurations, the apparent shape of similar 3D objects differs significantly in the 2D images. Bottom Left: **camera intrinsic**. The focal lengths and resolutions of the front cameras vary across the datasets, resulting in variations in the image plane. Notably, the Lyft dataset even includes two distinct focal lengths (highlighted in yellow). Bottom Middle: **longitudinal offset of cameras**. Bottom Right: **Ego Height in Different Datasets**

fails to achieve meaningful performance (1.1 mAP) when tested on Waymo. Our findings suggest an intriguing direction in 3D perception—how to train models on data from different sources.

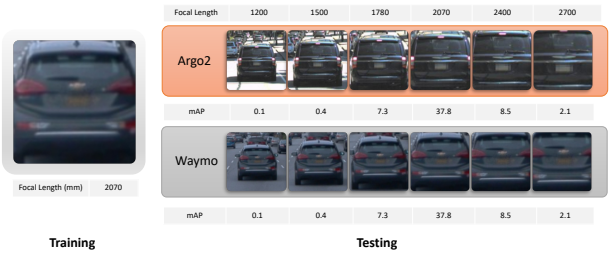


Figure 2: The detection performance on Argoverse2 and Waymo varies with different focal length settings. The highest mAP is achieved when the test-time focal length matches the focal length of the training data (*i.e.*, 2070mm), rather than using the original focal length values of 2070mm for Waymo and 1780mm for Argoverse2.

Therefore, the problem boils down to why incorporating additional data hampers model performance. What differences between the datasets are responsible for such catastrophic failures? We hypothesize that discrepancy in sensor parameters leads to these failures. We conduct a simple experiment to investigate this question. We train a model on Waymo and test the model on Argoverse2. We adjust the test image size and the focal length. When the focal length of Argoverse2 test images matches that of training images, the model achieves similar performance to Waymo test performance.

This observation further confirms our hypothesis that variations in camera parameters have a substantial impact on 3D detection. The underlying explanation can be found in the nature of imaging [12]: an image is a 2D projection that captures and renders visual information from the 3D physical world. As a result, varying sensor configurations leads to distinct projections, introducing geometric ambiguity in depth estimation. As depicted in Fig. 1, similar cars can exhibit significantly different apparent shapes in images.

In general, we refer to this issue as "*sensor misalignment*" across different datasets, and our extensive analysis confirms that parameters in the sensor suite (*i.e.*, intrinsic, extrinsic, and the ego coordinate system, illustrated in Fig. 1) play pivotal roles in the sensor misalignment (§ 3.4). To counter this misalignment, we propose simple approaches that utilize sensor parameters to account for biases in the input signals. First, we resize all the input images to synchronize the focal lengths. Second, an extrinsic aware module is deployed in the detection pipeline, mitigating the impact of camera translations. Third, ego frame alignment is applied to address the ambiguity in ego frame definition and the impact of camera height entangled with the ego center. Our method yields a massive improvement in the generalization capability of the detector, with an average precision increase of over 20 mAP on most datasets. In summary, the contributions of this paper are:

- We conduct a comprehensive evaluation to investigate the key issues that lead to performance drop in cross-dataset testing and multi-dataset training, analyzing how they affect 3D detectors. Our findings highlight three key factors: intrinsic, extrinsic, and ego coordinate system.
- We propose a simple yet effective sensor alignment method to address this issue by correcting the input signals. Through this correction, we demonstrate significant improvements in performance across all datasets.
- We achieve strong performance on a variety of datasets. Compared to direct transfer, our approach achieves an average improvement of more than 25 mAP on all datasets. Our joint training models achieve much better performance than any models trained on single dataset, and surpass individual models trained on Lyft, KITTI, and KITTI-360 datasets, even in the absence of these datasets during training.

2 Related Work

Camera-based 3D Detection. Recently, there has been significant progress in camera-based 3D object detection algorithms in the Bird's Eye View (BEV) space [4, 6, 13, 5, 14, 15, 7, 16, 17]. These algorithms have demonstrated strong performance and the ability to leverage information from multiple cameras and modalities to improve detection capabilities. These methods utilize camera intrinsic and extrinsic parameters to transform 2D image features into 3D space, a process known as BEV feature sampling. Depending on the sampling approach, there are two main categories: query-based methods and LLS(Lift-splat-shoot)-based methods [18]. Query-based [4, 13, 5, 8] methods utilize the object query mechanism of DETR [19] to generate 3D object queries. These queries are projected onto the image plane for feature sampling. On the other hand, LSS-based methods [6, 7, 16] predefined fixed 3D grid points and incorporate depth estimation modules to provide depth priors to the sampled features. In this paper, we will experiment with representative algorithms from these two categories, namely DETR3D [4] and BEVDet [6], as baselines.

Domain Adaptation. This line of methods aims to improve model performance from the source domain to the target domain[20, 21, 22, 23, 24, 25, 26, 27, 28]. Unlike these methods are trained on a single dataset, whereas our research covers the scenario of training on multiple datasets. This allows us to explore the challenges in a more diverse and realistic setting. Amongst, [29] and [30] are the studies related to us, which focus on domain adaptation in the context of 3D detection. In paper [29], the authors explore the impact of data distribution on the cross-dataset performance of LiDAR-based 3D detectors. And [30] primarily investigates the influence of camera intrinsic parameters on camera-based 3D detectors. Our research focuses on camera-based methods where the primary domain gap is sensor configuration rather than data distribution, and conduct our study is more comprehensive, including camera intrinsic parameters, camera extrinsic parameters, and ego coordinate system. We also avoid employing similar data augmentation techniques, as they can introduce biases to the data.

Multi-dataset training. There are several works have been shown effective for improving generalization capability by jointly training multi-datasets. In the field of monocular depth estimation, MiDAS [31] introduced five datasets for training and revealed the generalization power of mixing data from complementary sources. Uni3D [30] focus on joint training strategies and their impact on LiDAR-based 3D object detection. Both studies mentioned that naively adding datasets does not

Table 1: Dataset overview.

Dataset	#frame	Image resolution	Object type	#RGB camera
Argoverse2 (A)	26,687	(2048, 1550)	26	7
Kitti (K)	7,481	(1224, 370)	8	2
Kitti-360 (K360)	61,569	(1408, 376)	26	4 (2 fisheye)
Lyft (L)	22,680	(1920, 1080)/(1224, 1024)	9	7
nuScenes (N)	28,130	(1600, 900)	8	6
Waymo (W)	39,614	(1920, 1280)	3	5

unconditionally improve performance. We also encountered similar issue in the area of vision-centric 3D object detection, and we address it by employing cross-dataset sensor alignment.

3 Experiment and Analysis

3.1 Experiment Protocols

Datasets. We use six datasets in our experiments: Argoverse2 [2], KITTI [1], KITTI-360 [32], Lyft [33], nuScenes [11] and Waymo [3]. We focus on data related to camera-based 3D object detection. Each of these datasets offers ground-truth 3D bounding box labels for several kinds of objects. A comprehensive overview of the six datasets is provided in Table 1. Dataset format: Conducting cross-dataset analysis can be challenging due to the differences of data formats across datasets. To overcome this challenge, we leverage the unified format provided by MMDetection3D, which is compatible with all of our datasets. This allows us to merge and analyze the datasets seamlessly. For more detailed information of the conversion process and dataset settings, please refer to the Supplementary Materials.

Table 2: The random jittering experiments on the nuScenes dataset. We introduced a random translation within $[-2m, 2m]$ on the x or y coordinate for the ego center, in order to simulate jittering in camera extrinsic. The x-axis corresponds to the car's moving direction.

methods	None	on y	on x	on x,y
Direct	36.3	34.7	27.9	26.9
With EAM	36.9	35.5	35.4	34.9

3D object detection algorithms. We apply two camera-based models DETR3D [4] and BEVDet [6] to detect 3D objects in 2D images. DETR3D is a query-based detector that generates sparse 3D query points, projects them onto the 2D image, and samples corresponding features. On the other hand, BEVDet applies LSS(Lift Splat Shot) [18], a pillar-based view transform method, to project dense image feature back to a canonical 3D space. The projected features, commonly referred to as Bird's Eye View (BEV) features, are then utilized to predict object information. For model architecture, we use a Resnet-50 pretrained on ImageNet dataset as the image backbone. Following the original papers, the original resolution images are used as input for DETR3D, while for BEVDet, images at 1/4 resolution are utilized. During training, no augmentation techniques are applied except for photometric distortion. In the main paper, we focus on presenting results for single-camera detection. Additional experiments involving multi-view cameras and detailed training parameter settings can be found in the Supplementary Material.

Metric. Following the 2022 Waymo Open Dataset Challenge [3], we adopt the longitudinal error tolerant 3D average precision (LET-3D-AP) [34] as our evaluation metric. To unified taxonomy of all datasets, we merge the datasets' labels into three categories: vehicle, pedestrian, and bicycle. We report the LET-3D-AP of each class using Intersection over Union (IoU) thresholds at 0.5, 0.3 and 0.3 following the Waymo settings. We filter the objects within a range of 51.2 meters for both training and testing. For simplicity, we report average the AP scores of each class (mAP) in the main paper. To ensure result consistency, the experiments and discussions in this paper are centered around the outcomes of DETR3D, unless stated otherwise. We additionally perform similar experiments using BEVDet to validate the generalizability of our findings to other detector types and report the results of BEVDet in Supplementary Material.

3.2 Train on one dataset and test across datasets

We initially evaluate the performance of camera-based detectors on individual datasets to assess their learning capability and then examine their performance across datasets and summarized the results in Table 3. Regarding within-domain performance, DETR3D exhibits satisfactory results on Waymo and Argoverse2 (AV2). However, the performance declines on Lyft and nuScenes datasets. The Model performs worst on KITTI and KITTI-360 dataset. One cause of these variations is diverse data collection condition. Lyft dataset exhibits diversity in sensor settings as it was collected by 20 different autonomous vehicles, while nuScenes is diverse in location, including data from two

different continent (*i.e.*, Singapore and USA). Besides, Insufficient data volume and unbalance data distribution also attribute to these difference. The KITTI dataset only have 4,000 training samples which hampers model convergence. The lack of pedestrian and bicycle annotations in KITTI-360 results in a drastic drop in the average performance across all three classes. Regarding cross-domain performance, we find that the mAP almost drop to 0 for most dataset pairs, except for the pair between Waymo and AV2. The failure cannot be attributed to distribution shift or label space discrepancies, as the detector also faced challenges when transferring knowledge between KITTI and KITTI-360, which are collected within the same city during a similar timeframe. We further conducted a simple experiment on the Waymo dataset by modifying the camera intrinsic parameters and scaling the images accordingly, followed by retesting. Surprisingly, in Figure 2 we observe that changes in resolution has a significant impact on the model's performance, and the model performs as poor in Waymo itself as in AV2 when the image is resized to the focal length of AV2. On the other hand, the performance on AV2 greatly improve when we resize the images in AV2 to Waymo's focal length.

This inspire us that camera intrinsic may be the root cause of performance drop across dataset, as different focal lengths introduce deviations in images, which are crucial for camera-based detectors. In fact, the focal lengths is quite different accross datasets, as shown in Fig. 1. In next section, we attempt to address this issue by incorporating more training data.

3.3 Train on multiple datasets and test across datasets

To improve the data diversity, We progressively incorporated the nuscenes, AV2, Lyft, KITTI, KITTI-360, and Waymo datasets into the training set, and train six models[1]. These models are then evaluated on all six datasets, and the results are presented in the first block of Table 4. There are some improvements as more data became available. For example, the mAP for nuScenes gradually increases from 36.3% to 46.2%. But there is also a clear drop in mAP (from 41.4% to 36.3%) for KITTI, when KITTI-360 was added to the training set. Moreover, despite including more data for training, the models still struggle to detect objects in the unseen datasets, suggesting that simply adding training data does not improve generalization. In the following sections, we will model the 3D-2D correspondence of objects and formally analyze how 3D detectors predict objects based on 2D images, in order to gain a better understanding of how focal length affects detection performance. Apart from camera intrinsic, we also discover the impact of camera extrinsic and ego coordinate system on the detection performance.

3.4 Analysis

3D-2D correspondence of objects. Experiments of last two sections reveal the limitation of current image-based detectors in generalization: the deviations in dataset cannot be easily learned by the detectors, even with the inclusion of multiple datasets. Figure 1 demonstrates the impact of this deviation: similar 3D information corresponds to highly distinct 2D image information. So, how do the dataset gap contribute to the detection failure?

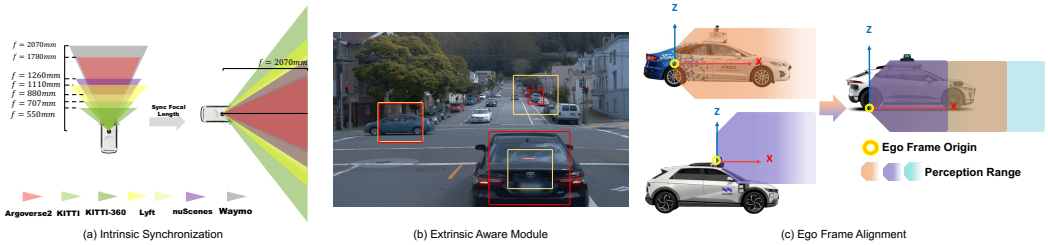
We first establish a pinhole camera model to investigate the differences. Assuming a frontal camera with focal length f_x located at (t_x, t_y, t_z), and a car locating at (x, y, z) in ego frame, with real-world size S and pixel size s_{pixel}, their relation can be formulated as:

$$s_{pixel} = f_x \times \frac{S}{x - t_x}, \tag{1}$$

where $x - t_x$ represents the depth in the camera frame. According to the equation, focal length f_x and camera location t_x is changing, the same car in the same location will look differently, and this is the exact challenge we face during the process of cross-dataset training and testing (Fig. 1 and Fig. 1).

3D detection pipeline. To explore this question further, we examine the detection process of a specific object: the 3D detector first utilizes camera intrinsic and extrinsic parameters to project a 3D query point p_0 to a 2D position (u,v) to sample feature on the image plane. The sampled feature, together with the prior knowledge at p_0 is then used to make an object prediction. This process can

[1]From our experiment, the trend in performance is consistent regardless of the order in which the datasets are added. For the simplicity, we stick with this setting throughout the main paper.



(a) Intrinsic Synchronization (b) Extrinsic Aware Module (c) Ego Frame Alignment

Figure 3: **(a)**: the Camera Intrinsic is synchronized by resizing the images. **(b)**: Camera Extrinsic impact is decoupled by Extrinsic Aware Module (EAM). Yellow rectangles represent the original feature sample's recipient field, while the red rectangles represent the recipient field of the model with EAM. **(c)** Ego frames aligned by shifting the ego center in each datasets.

be represented by:

$$\mathcal{H} : (I(u-s, v-s, u+s, v+s), p_0) \longrightarrow \hat{\mathbf{b}}_0, \hat{c}_0, \tag{2}$$

where $I(u-s, v-s, u+s, v+s)$ represents a patch of the image I centered at (u, v) with a size of $2s \times 2s$. This patch corresponds to the image feature sampled from point (u,v). $\hat{\mathbf{b}}_0$ represents the object's position and size, while $\hat{c}_0$ represents the confidence of the object belonging to a certain class. Additionally, according to the pinhole camera model, the projection relationship between p_0 and (u, v) can be expressed by the following equation:

$$d(u, v, 1) = KTp_0, \tag{3}$$

where K and T represent the camera intrinsic and extrinsic matrices, respectively, and d represents the depth of p_0 in the camera coordinate system, which is contained in the expression KTp_0. During the training phase, the detector learns to predict an object based on the input pair: p_0 and its corresponding 2D image patch $I(u-s, v-s, u+s, v+s)$.

By examining Eqs. (1) to (3), we can find that when considering the same values for $\hat{\mathbf{b}}_0$ and $\hat{c}_0$, the differences in intrinsic K, extrinsic T introduce variations to the corresponding image patch $I(u-s, v-s, u+s, v+s)$ between datasets. Furthermore, the coordinate of query p_0 depends on the ego coordinate system, which varies a lot in different datasets (Fig. 1). This would change the distribution of the 3D query points, deviating the inherent prior knowledge. For example, the bottom of an object is in contact with the ground and the distribution of bottom heights can be assumed as a normal distribution, denoted as $N(0, \sigma)$. In the ego frame, the z-coordinate of the object's bottom follows $N(-h, \sigma)$, where h represents the height of the ego center. If the detector is trained on dataset A with ego height h_A, but tested on dataset B with ego height h_B, it would suffer from a distribution shift from $N(-h_A, \sigma)$ to $N(-h_B, \sigma)$. In other words, using different sensor configurations to observe the same vehicle results in highly inconsistent mapping relationships $\mathcal{H}$. During cross-dataset evaluation, a mismatch between the image patch and p_0 arises, leading to incorrect predictions. In summary, the sensor deviation between datasets is three-folds:

- **Intrinsic.** There is variation in camera intrinsic parameters, specifically, the focal length (Figure 1). The same car at the same location appears differently in the images when we switch datasets.
- **Extrinsic.** Apart from the f_x, the extrinsic parameters or camera poses t_x can also affect the object size in the image from the Eq. (1). we visualized the difference of camera poses across datasets in the left of Figure 1.
- **Ego frame coordination.** The variation in the height of ego centers introduces inconsistency in the data distribution. As illustrated in right of Figure 1, the value of h differs significantly across datasets, thus leading to unreliable query prior information during cross-dataset testing.

4 Sensor Alignment Approaches

We employ three simple yet efficient strategies in both the training and testing schemes to address the three challenges posed by sensor deviation correspondingly. The idea of these strategies is to establish a dataset-invariant 3D-2D mapping relationship, allowing the models to perform consistently across different datasets.

Table 3: Cross-dataset testing results of DETR3D[4] trained on single dataset. We report the average LET-3D-AP[34] for vehicles, pedestrians, and bicycles. Each row represents the source domain, and each column represents the target domain. The **bold** font highlights the within-domain results, (i.e., training and testing using the same dataset). The detector suffers a significant performance drop when applied directly to unseen datasets (first block). With the aid of intrinsic synchronization, the average performance on all datasets improves in the second block. DETR3D achieves its best results in the third block.

Setting	src/dst	N	A	L	K	K360	W	avg.
Direct	N	**36.3**	0.8	1.8	0.0	0.0	1.1	6.7
	A	0.2	**48.0**	0.1	0.0	0.0	17.4	11.0
	L	0.5	0.1	**37.3**	0.4	0.0	0.1	6.4
	K	2.8	1.2	0.0	**24.5**	1.1	0.7	5.0
	K360	0.1	0.2	0.0	3.2	**26.1**	0.1	5.0
	W	0.1	8.9	0.0	0.0	0.0	**58.8**	11.3
Sync Intrinsic	N	**35.7**	21.9	13.8	27.1	16.9	19.2	22.4
	A	11.8	**46.8**	7.2	6.4	5.4	38.4	19.3
	L(fx2070)	1.0	1.3	**44.0**	8.1	5.7	1.5	10.3
	K	1.2	0.2	0.6	**24.7**	6.4	0.1	5.5
	K360(fx2070)	14.6	14.7	7.3	34.6	**34.7**	8.2	19.0
	W	14.5	37.8	14.3	9.4	5.6	**57.7**	23.2
Sync Extrinsic and Ego center	N	**43.1**	33.6	32.8	33.0	18.4	33.0	32.3
	A	24.4	**48.1**	34.1	18.1	8.7	37.4	28.5
	L	15.7	19.6	**47.1**	20.0	12.9	18.9	22.4
	K	7.1	8.7	10.2	**29.1**	9.3	2.4	11.1
	K360	13.9	17.7	16.6	39.1	**36.7**	8.4	22.1
	W	25.4	38.2	33.6	21.2	11.7	**57.6**	31.3

Table 4: Multi-dataset training results by DETR3D (evaluated on the validation sets). Beginning with nuScenes, We gradually incorporate the training splits of six datasets . Directly merging datasets for training yields no improvement for cross-dataset testing. However, with sensor alignment approaches, DETR3D benefits from the increasing data volume, and performs much better in both within-domain and cross-domain evaluation.

Setting	src/dst	N	A	L	K	K360	W	avg.
Direct	N	36.3	0.8	1.8	0.0	0.0	1.1	6.7
	+A	40.5	49.2	0.5	0.0	0.0	5.2	15.9
	+L	41.6	50.5	43.7	0.0	0.0	3.8	23.3
	+K	41.5	49.7	46.0	41.4	1.1	3.6	30.5
	+K360	42.6	54.3	46.8	36.3	29.7	3.3	35.5
	+W	46.2	53.7	49.4	39.5	29.7	61.9	46.7
Sync Intrinsic	N	40.8	25.5	18.6	29.7	18.0	23.4	26.0
	+A	45.5	50.0	25.1	35.8	21.3	44.2	37.0
	+L	46.8	53.2	55.1	37.8	23.1	45.3	43.5
	+K	47.4	53.5	53.6	57.8	21.8	44.4	46.4
	+K360	50.2	54.4	54.0	60.2	39.6	44.7	50.5
	+W	51.8	55.3	56.6	61.9	40.7	63.7	55.0
Sync Extrinsic and Ego center	N	43.1	33.6	32.8	33.0	18.4	33.0	32.3
	+A	52.1	52.7	38.4	42.2	23.2	40.7	41.6
	+L	52.6	53.2	59.5	46.1	26.1	43.6	46.9
	+K	51.0	54.7	60.2	63.9	28.4	44.6	50.5
	+K360	50.0	55.0	59.8	65.0	42.7	45.2	52.9
	+W	54.8	56.4	60.5	66.8	43.4	62.7	57.4

4.1 Intrinsic Synchronization

Camera intrinsic is the easiest and most important factor to synchronize. Inspired by intrinsic-decoupled method commonly used in depth estimation[30], we propose a straightforward strategy of resizing the images to a fixed focal length, denoted as f_0, using bi-linear interpolation. This strategy, as depicted in Figure 3, ensures that the focal length remains consistent across different datasets. Specifically, we synchronize the focal length of the target domain to match that of the source domain (*i.e.*, the training data) during cross-domain evaluation, and we synchronize all focal lengths to be the same as Waymo's, which has the largest focal length, in multi-dataset training.

As shown in Table 3, simply resizing the images can significantly improve the LET-3D-AP. The model trained on nuScenes shows an improvement in cross-domain performance of more than 20% on Argoverse2 and KITTI. Similarly, the model trained on KITTI-360 achieves a mAP of 34.6% on the KITTI dataset, which is almost the same as the performance of KITTI-360 itself. This aligns with the fact that the two datasets have minimal domain gap except for the difference in focal length (707mm vs. 552mm). The results from the multi-dataset settings in Table 4 further demonstrate the potential of this approach. Both within-domain and cross-domain performances are significantly enhanced, and the negative effect of KITTI-360 observed in the previous section is eliminated. This suggests that the model indeed benefits from an increase in data volume.

4.2 Extrinsic Aware Module (EAM)

The deviation in camera extrinsic, specifically, camera frontal translation t_x, cannot be addressed through operations on images as they introduce bias. For example, warping image by a homography matrix to simulated changes in camera extrinsic[30]. To tackle this issue, we introduce a camera-extrinsic-aware module during the view transformation between $p_0(x_0, y_0, z_0)$ and (u, v) of Eq. (3). According to Eq. (1), t_x affects the apparent size of objects, while in Eq. (2), for a 3D query point p_0 in the detector, its perception field in the image is a fixed value of $2s$. This means that the detector is unaware of the change in image brought by t_x. To address this, we enlarge the perception field of p_0 to $2s' = 2s * \frac{c}{x_0 - t_x}$, where c is a constant. As a result, the input pair of the image patch and p_0 becomes invariant to variations in t_x. This operation is achieved by sampling more points and applying average pooling, similar to ROI Align [35].

We apply random translations to the ego center to simulate changes in camera position to evaluate the robustness of our approach. Table 2 demonstrates that our proposed method is more robust to jittering, indicating the effectiveness of the extrinsic aware module. We also observe that t_y has a negligible impact. As for t_z, it will be addressed in the next approach, along with the ego center.

We will also conduct cross-dataset testing and multi-dataset training in combination with the next approach.

4.3 Ego Frame Alignment

The definition of the ego frame varies across datasets, with some being higher than 1.5 meters, which can lead to variations in the data distribution, leading to unreliable query prior in Eq. (2). We first transform the ego coordinate systems in all dataset to in contact with the ground (i.e. ego height is 0m). In this way, we can make sure the z-coordinate represent the same physical meaning across different datasets. However, we observe that, naively training and testing with this operation leads to no improvement. We further incorporate intrinsic synchronization and search for the best alignment strategy. In Fig. 4, we modify the x and z coordinates of the ego center for each dataset and evaluate them using a model trained on Waymo. We identify the brightest cells in each dataset as the optimal ego settings relative to Waymo's ego center. Interestingly, we observe that the model's performance is not solely affected by the height of the ego center but also by its frontal position. We attribute this phenomenon to the distribution of object x-coordinates in each dataset. This cannot be resolved by the extrinsic aware module, which is used for 3D-2D mapping, rather than 3D distribution. As depicted in Figure 3, we use the optimal ego translation parameters to align all the ego coordinate systems. By selecting the optimal ego settings in conjunction with the aforementioned strategies, we achieve even better results in both cross-dataset testing and multi-dataset training. As shown in Table 4 and 3, compared to direct merging, the performance across all six datasets is significantly improved, with gains of up to 27.4% (KITTI).

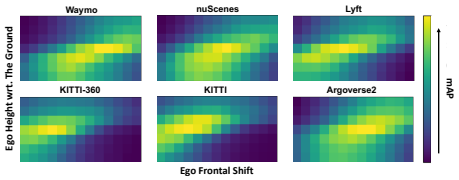


Figure 4: Detector performance vs. ego center translation in x and z coordinates. We train DETR3D on Waymo and test it with different ego frame translations on all datasets. Each grid point in the heatmap represent $0.5m \times 0.5m$. The brightest cells in each heatmap represent the optimal ego center settings of each dataset.

5 Conclusion

In this paper, we conducted a comprehensive analysis of the challenges faced by image-based detectors in achieving high-performance and generalization capabilities across multiple autonomous driving datasets. We identified the inconsistent 3D-2D mapping relationship as a dominant factor, resulting from variations in sensor configuration including camera intrinsic, camera extrinsic, and ego coordinate system. Surprisingly, we found that simple sensor alignment operations significantly mitigate the performance drop. In cross-dataset testing, we achieve an average performance boost of 25.6 mAP from nuScenes to all datasets, fully enjoying the advantage of highly diverse data distribution in nuScenes. We also achieve 17.5, 16.0, 6.1, 17.1, 20 mAP boost when AV2, Lyft, KITTI, KITTI-360 and Waymo serve as the source domain, respectively. It is worth noting that our metric considers objects of three classes: vehicle, pedestrian and bicycle, while most studies often ignore the latter two. In multi-dataset training, we fully exploit the potential of data volume, with 18.5, 8.4, 23.2, 42.3, 17.3, 3.9 mAP gaining by combining six datasets instead of training on them separately. Compared to direct merging, we achieve an average performance boost of more than 10 mAP on all datasets. We believe that our findings will stimulate further research in multi-dataset training and domain adaptation for vision-centric 3D object detection and localization. We emphasize the importance of applying simple data corrections before incorporating additional datasets or developing new computer vision algorithms to bridge the remaining domain gaps.

6 Limitations.

While our study provides valuable insights into addressing sensor misalignment in image-based detectors for autonomous driving, there are two limitations to consider: Due to the lack of autonomous driving datasets with highly diverse camera poses, we were unable to study the impact of camera rotation on the performance of the detection models. We were unable to scale our method to a larger-scale in-the-wild dataset due to annotation scarcity. This may be alleviate by using the semi-supervised 3D detector as the baseline, and we leave it to the future work.

References

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[2] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

[3] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. 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.

[4] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.

[5] Y. Liu, T. Wang, X. Zhang, and J. Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 531–548. Springer, 2022.

[6] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.

[7] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022.

[8] 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. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 1–18. Springer, 2022.

[9] J. Park, C. Xu, S. Yang, K. Keutzer, K. Kitani, M. Tomizuka, and W. Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. *arXiv preprint arXiv:2210.02443*, 2022.

[10] X. Lin, T. Lin, Z. Pei, L. Huang, and Z. Su. Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. *arXiv preprint arXiv:2211.10581*, 2022.

[11] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[12] J. T. Wixted and S. L. Thompson-Schill. *Stevens' Handbook of Experimental Psychology and Cognitive Neuroscience, Language and Thought*, volume 3. John Wiley & Sons, 2018.

[13] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao. Futr3d: A unified sensor fusion framework for 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 172–181, 2023.

[14] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, and Z. He. A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[15] Y. Liu, Y. Wang, Y. Wang, and H. Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2022.

[16] Y. Li, H. Bao, Z. Ge, J. Yang, J. Sun, and Z. Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022.

[17] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023.

[18] J. Philion and S. Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020.

[19] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[20] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[21] C.-D. Xu, X.-R. Zhao, X. Jin, and X.-S. Wei. Exploring categorical regularization for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11724–11733, 2020.

[22] Z. He and L. Zhang. Domain adaptive object detection via asymmetric tri-way faster-rcnn. In *European conference on computer vision*, pages 309–324. Springer, 2020.

[23] G. Zhao, G. Li, R. Xu, and L. Lin. Collaborative training between region proposal localization and classification for domain adaptive object detection. In *European Conference on Computer Vision*, pages 86–102. Springer, 2020.

[24] D. Acuna, J. Philion, and S. Fidler. Towards optimal strategies for training self-driving perception models in simulation. *Advances in Neural Information Processing Systems*, 34:1686–1699, 2021.

[25] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[26] H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018.

[27] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera. Cam-convs: Camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11826–11835, 2019.

[29] Y. Wang, X. Chen, Y. You, L. E. Li, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020.

[30] S. Wang, X. Zhao, H.-M. Xu, Z. Chen, D. Yu, J. Chang, Z. Yang, and F. Zhao. Towards domain generalization for multi-view 3d object detection in bird-eye-view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13333–13342, 2023.

[31] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

[32] Y. Liao, J. Xie, and A. Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[33] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, et al. Lyft level 5 av dataset 2019. *urlhttps://level5. lyft. com/dataset*, 1:3, 2019.

[34] W.-C. Hung, H. Kretzschmar, V. Casser, J.-J. Hwang, and D. Anguelov. Let-3d-ap: Longitudinal error tolerant 3d average precision for camera-only 3d detection. *arXiv preprint arXiv:2206.07705*, 2022.

[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.