Understanding Cross-Rig Generalization in Automotive Perception: a Multi-Rig Benchmark and Rig Variation Metrics

Tim Alexander Bader[1,2], Tim Dieter Eberhardt[1,2], Maximilian Dillitzer[1,3], and Wilhelm Stork[2]

[1] Dept. of Highly Automated and Assisted Driving, Dr. Ing. h.c. F. Porsche AG, Porscheplatz 1, Stuttgart, 70435, Baden-Württemberg, Germany
{tim.bader1,tim.eberhardt1,maximilian.dillitzer}@porsche.de
[2] Institute for Information Processing Technologies, Karlsruhe Institute of Technology, Kaiserstraße 12, Karlsruhe, 76131, Baden-Württemberg, Germany
[3] Faculty of Mobility and Technology, Esslingen University of Applied Sciences, Kanalstraße 33, 73728 Esslingen am Neckar Esslingen, Germany

Abstract. Camera-based perception systems for autonomous driving are typically developed and evaluated using fixed sensor rigs, while real-world vehicle fleets exhibit substantial variation in camera placement, orientation, field of view, and camera count. This mismatch introduces a cross-rig domain gap in which only the geometric observation process changes. To study this effect under controlled conditions, we introduce *Plentiful CARLA Camera Rigs*, a benchmark that renders identical driving scenes under 14 systematically designed camera rigs. This setup enables direct analysis of cross-rig generalization without confounding changes in scene content or appearance. Using the benchmark, we analyze cross-rig transfer behavior of representative multi-view perception architectures and observe substantial performance shifts induced by geometric rig variation. To facilitate structured analysis, we further introduce two calibration-based descriptors derived from rig metadata: *Rig Variance*, capturing internal rig diversity, and *Rig Contrastive Distance*, measuring geometric discrepancy between rigs. Our experiments show that geometric rig differences strongly correlate with relative cross-rig performance shifts and that Rig Contrastive Distance provides a reliable proxy for ranking transfer difficulty between sensor rigs.

1 Introduction

Camera-based perception systems for autonomous driving have achieved strong performance under carefully engineered and largely fixed sensor configurations. In other words, during both training and evaluation, a specific number of cameras with fixed positions, orientations, and fields of view is required [2,29,34]. In practice, however, vehicle fleets are heterogeneous. Differences in vehicle segment, packaging constraints, cost targets, and generational updates lead to variations in camera placement, overlap, field of view (FOV), and even camera count.

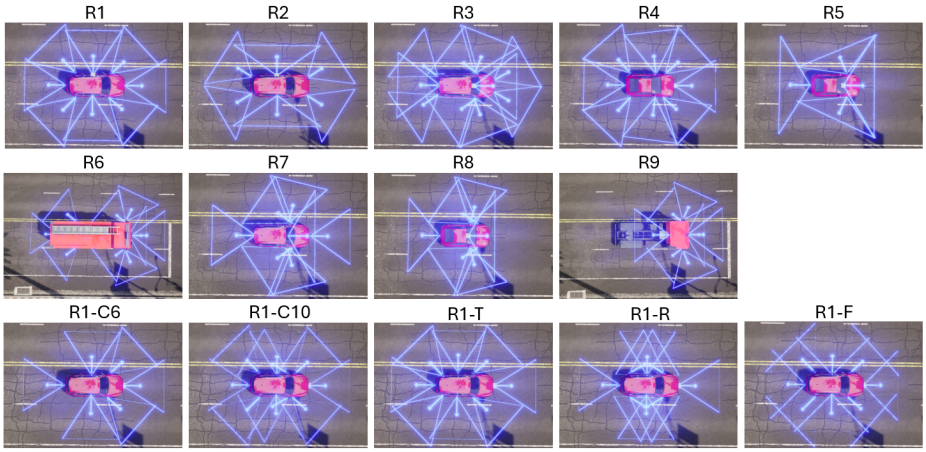

Fig. 1: The rigs of our Plentiful Carla Camera Rigs benchmark. It contains nine unique rigs (top/center) and five factor-modified control rigs of R1 (bottom). Shown are the camera FOVs and the view directions.

This setting raises a fundamental question: *how do camera rig characteristics alone influence perception performance?* While this problem is often implicitly absorbed into broader notions of domain shift, cross-rig transfer differs in an important way: only the geometric observation process varies. Performance shifts therefore arise purely from differences in viewpoint distribution, coverage, and redundancy. We refer to this phenomenon as the *cross-rig domain gap*.

The idea of decoupling perception models from specific sensor rigs has received only limited attention in prior work [1, 32]. What exists is Visual Question Answering (VQA) across two datasets with different sensor rigs [33] and the observation of a *cross-sensor domain gap* [6, 13]. However, most benchmarks typically assume fixed sensor configurations [1, 20] or limited viewpoint perturbations [6, 13], and cross-dataset evaluations conflate rig changes with scene statistics, appearance differences, and dataset bias. As a result, there is currently no comprehensive protocol to disentangle performance variation caused specifically by rig characteristics.

To address this limitation, we introduce *Plentiful Carla Camera Rigs*, a benchmark in which identical scenes are rendered under 14 systematically designed camera rigs, shown in Figure 1. By keeping scene content, object layout, and environmental conditions fixed while varying only calibration parameters, the benchmark isolates rig-induced geometric effects. Importantly, this setting does not require modeling camera-specific photometric characteristics or hardware artifacts. Because our goal is to study *relative performance shifts induced by geometric observation changes*, a high-fidelity simulator such as CARLA is sufficient: it allows controlled manipulation of extrinsics and intrinsics while holding all other factors constant.

This controlled benchmark enables systematic analysis of how camera rig geometry influences perception performance. To facilitate such analysis, we further introduce two rig-centric descriptors derived from calibration metadata. First, *Rig Variance* summarizes the internal geometric diversity and overlap structure of a single rig. Second, *Rig Contrastive Distance* measures geometric discrepancy between two rigs in terms of camera placement, orientation, FOV and coverage. They provide a structured way to reason about how much the viewing geometry changes when transferring between rigs. We empirically evaluate the relationship between these geometric descriptors and observed performance shifts across multiple camera-based 3D perception architectures.

Our contributions are summarized as follows:

- We introduce *Plentiful Carla Camera Rigs*, a controlled benchmark featuring identical scenes rendered under 14 systematically varied camera rigs.
- We propose two calibration-metadata-based geometric descriptors, Rig Variance and Rig Contrastive Distance, for quantifying rig properties and cross-rig discrepancies independent of any specific perception model.
- We provide empirical analysis demonstrating that geometric rig differences systematically relate to relative cross-rig performance shifts.

Any relevant supplementary information can be accessed through our project page at `https://badertim.github.io/plentiful-carla-camera-rigs/`.

2 Related Work

We highlight existing work about rig descriptors, followed by rig-agnostic architectures and relevant data resources.

2.1 Rig Configuration and Viewpoint Diversity

Existing work on sensor placement for autonomous vehicles primarily evaluates configurations through task-dependent performance metrics such as coverage [12], detection accuracy [37], or information gain [38]. A prominent line of research introduces *perception entropy* [7, 25], which quantifies the expected uncertainty reduction provided by a given multi-sensor rig and has been applied to both camera–LiDAR configurations and optimization of roadside sensors. Closely related LiDAR-centric studies analyze non-detectable subspaces using geometric measures such as the volume-to-surface-area ratio (VSR) [23], or propose information-theoretic surrogates to predict 3D detection performance under different placements [9, 17]. While these approaches demonstrate the strong impact of configuration on downstream perception, they are inherently tied to specific models, datasets, or environment assumptions.

A complementary body of work focuses on *viewpoint quality* and *view diversity*, most notably through viewpoint entropy [16, 35, 36] to power several downstream tasks using optimized view selection. These methods capture how informative individual viewpoints are but do not provide a unified, rig-level descriptor for multi-camera systems.

2.2 Rig-Agnostic Perception Architectures

To overcome fixed-sensor constraints, recent work has focused on decoupling 2D feature extraction from 3D geometric reasoning.

Geometric Projection and Lifting. Lift-Splat-Shoot (LSS) [31] introduced explicit view transformation by lifting image features into 3D frustums and projecting them onto a BEV grid via calibration, enabling geometric consistency across camera rigs. Building on this idea, *BEVDet* [10] improves robustness through independent image and BEV augmentations, while *BEVFusion* [19] extends the approach to camera–LiDAR fusion. The *Detecting As Labeling* (DAL) [11] framework further mitigates overfitting by separating candidate generation and regression across modalities.

Query-Based and Implicit Representations. Alternative approaches use sparse queries or implicit encodings to connect 2D observations and 3D reasoning. *DETR3D* [41] projects 3D object queries onto image features, while *PETR* [21, 22] injects 3D positional embeddings into image features to enable implicit geometric reasoning. Architectures such as *BEVFormer* [18, 42] instead rely on learned spatial cross-attention, which can be more sensitive to rig changes.

Unconstrained Multi-View Reconstruction. Another direction explores unconstrained multi-view reconstruction from arbitrary image collections. Methods such as DUSt3R [40], VGGT [39], and Fast3R [43] demonstrate strong geometric reasoning under highly variable camera layouts. However, these approaches focus on scene reconstruction rather than semantic perception. Automotive-specific work such as Rig3R [14] conditions reconstruction on rig information but remains closed-source and difficult to evaluate reproducibly.

2.3 Data and Benchmarks

Progress in automotive perception has been closely tied to large datasets and benchmarks. Most 3D object detection methods [26] are evaluated on datasets such as nuScenes [2] and the Waymo Open Dataset [34], which provide diverse driving scenarios together with fixed sensor configurations. While these benchmarks are essential for comparing perception architectures using metrics such as mean Average Precision (mAP) and the nuScenes Detection Score (NDS), their static sensor rigs prevent controlled analysis of how changes in camera placement, orientation, FOV, or camera count affect perception performance.

Simulation environments provide a promising alternative, as they allow explicit control over sensor calibration and scene generation. The CARLA simulator [5] enables rendering identical scenes under different sensor configurations while keeping all other factors constant. Using this capability, Embacher et al. [6] introduced a dataset in which an initial rig was fitted to two different vehicle types, demonstrating the existence of a cross-rig domain gap. Related work has benchmarked viewpoint robustness by perturbing the cameras of an initial rig

according to several viewpoint factors [13]. However, existing efforts remain limited in scale and do not provide a systematic benchmark covering a broader spectrum rig variations.

Recent advances in simulation and synthetic data generation further expand the potential for controlled perception benchmarks [4, 30]. Frameworks such as NVIDIA Cosmos [27] and AlpaSim [28] enable scalable synthetic data generation but typically focus on increasing scene diversity rather than systematically varying sensor configurations.

Overall, existing datasets and simulation platforms either rely on fixed sensor rigs or provide unstructured synthetic diversity, limiting the study of rig-induced performance variation and motivating controlled multi-rig benchmarks.

3 Plentiful CARLA Camera Rigs Benchmark

The goal of the Plentiful CARLA Camera Rigs benchmark is to enable controlled analysis of cross-rig generalization in automotive perception. To isolate the effect of camera rig geometry, we render identical driving scenes under multiple systematically designed sensor configurations while keeping all other factors constant. This setup allows direct comparison of perception performance across rigs and enables reproducible study of how changes in camera placement, orientation, FOV, and camera count influence downstream models.

3.1 Data Generation Process

The dataset is generated with the CARLA Simulator [5] 0.9.16 (+ map expansion) using a two-stage simulation pipeline to ensure reproducibility and controlled variation across camera rigs. We first define a global metadata configuration specifying vehicle and sensor rigs, maps, weather conditions, dataset splits, and simulation frame rate (shown in Table 1 & 2). Based on this configuration, we randomly sample scene descriptors for the training, validation, test and mini splits. Each scene descriptor specifies map, weather condition, traffic density, and ego-vehicle spawn point. Splits are defined at the scene level to prevent data leakage.

For each scene descriptor, the simulator is initialized and populated with dynamic traffic using the traffic manager, while the ego vehicle operates in autopilot mode. As the default traffic manager is non-deterministic, we record full trajectories of all simulated entities in an initial pass. Trajectory recording is performed using the largest vehicle model used in the dataset to ensure compatibility across sensor configurations with different vehicles.

Before replay, we apply a trajectory-level pruning step to reduce the overrepresentation of near-stationary ego-vehicle behavior by computing per-scene mean ego speed, binning scenes by speed, and downsampling over-populated bins. The initial pool contains 100 trainval scenes and 40 test scenes, and pruning is configured to keep 80 trainval and 30 test scenes.

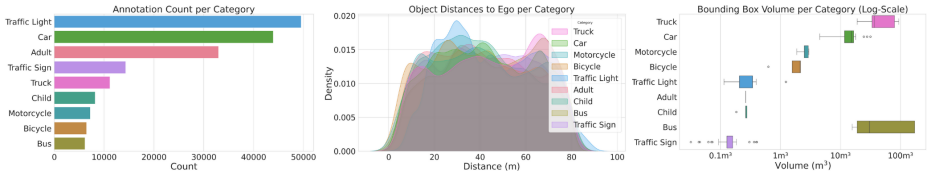

Fig. 2: Benchmark distribution of all accumulated splits of R1-c10. Shown are the category distribution (left), the object distance distribution to the ego vehicle (center) and the 3D bounding box volume distribution (right). The distribution clusters of the latter can be attributed to the different category types.

Next, all the kept scenes are replayed deterministically using the recorded trajectories. Each vehicle and sensor rig replays the same set of scenes, and the corresponding sensor data is recorded. This guarantees that different camera rigs observe identical scene dynamics.

Finally, all sensor data, 3D object annotations, and ego-vehicle metadata are stored in a standardized format inspired by nuScenes [2]. Dependent on the target format, the data is processed further to fit MMDetection3D [3].

3.2 Benchmark Statistics

The benchmark contains 115 scenes in total, where each has a fixed duration of 30 s and is recorded at 2 Hz, resulting in 60 samples per scene. Across all splits, this yields 6,900 samples. Considering the 105 cameras over all 14 rigs shown in Table 2 and Figure 1, this leads to 724,500 images in total.

The dataset provides 3D annotations for nine object categories: car, truck, bus, motorcycle, bicycle, adult, child, traffic sign, and traffic light. In the case of R1-c10, the benchmark contains 179,926 annotated objects corresponding to 7,299 unique instances. Annotations include oriented 3D bounding boxes, semantic attributes, and discrete visibility labels that distinguish between 99,308 fully visible, 18,316 mostly visible and 42,283 partially visible objects dependent on the pixel coverage of the cameras. This leads to minor annotation differences between rigs (e.g., R1-c6 has 159,907 annotations) to prevent ill-posed labels. The annotations are limited up to a range of 80 m from the ego-vehicle origin and all camera images are rendered at a fixed resolution of 1280×720 pixels.

The ego vehicle operates in autopilot mode and exhibits a diverse speed profile across scenes. Over the full benchmark, ego-vehicle speeds range from 0 to 47.99 km/h, with a mean speed of 16.02 km/h and significant distribution peaks at 0 km/h and 30 km/h. Table 1 shows more detailed information.

4 Rig Variation Metrics

Camera rigs for autonomous driving vary in placement, orientation, and FOV, and such variations strongly influence model performance. We introduce two metrics that quantify these differences based on camera poses and intrinsics.

Split	Scenes	Samples	Annotations	Maps	Weathers	Traffic Density
Train+Val	80	4,800	124,402	5	13	[0.2, 0.8]
Test	30	1,800	47,843	3	12	[0.1, 0.9]
Mini	5	300	7,681	2	2	[0.3, 0.7]
Combined	115	6,900	179,926	8	13	[0.1, 0.9]

Table 1: Dataset statistics summarizing scene count, sample count, object annotations, environmental diversity, and traffic density coverage over data splits. Annotation count slightly varies between rigs due to camera placement, shown here is R1-c10.

The metrics are complementary: Rig Variance characterizes the internal heterogeneity of a single rig, while Rig Contrastive Distance quantifies the discrepancy between two distinct rigs. Together, they enable a preliminary analysis of how changes such as translation shifts, rotational perturbations, or FOV modifications affect perception models.

4.1 Rig Variance

Rig Variance (RigV) measures the internal heterogeneity of a single rig and captures how diverse the cameras are in their spatial and optical configuration. Each camera i is represented as $\mathbf{c}_i = (\mathbf{t}_i, \mathbf{r}_i, f_i)$, where $\mathbf{t}_i \in \mathbb{R}^3$ denotes the position in ego coordinates, $\mathbf{r}_i \in SO(3)$ denotes the orientation, and f_i denotes the FOV RigV computes the average pairwise divergence within a rig

$$\text{RigV} = \frac{1}{N(N-1)} \sum_{i \neq j} D(\mathbf{c}_i, \mathbf{c}_j), \tag{1}$$

with

$$D(\mathbf{c}_i, \mathbf{c}_j) = \lambda_t \, \Delta t_{ij} + \lambda_r \, \Delta r_{ij} + \lambda_f \, \Delta f_{ij}. \tag{2}$$

Here, the individual components are defined as

$$\Delta t_{ij} = \frac{\|\mathbf{t}_i - \mathbf{t}_j\|_2}{D_{\max}}, \quad \Delta r_{ij} = \frac{d_R(\mathbf{r}_i, \mathbf{r}_j)}{\pi}, \quad \Delta f_{ij} = \frac{|f_i - f_j|}{180} \tag{3}$$

where d_R denotes the geodesic rotation distance on $SO(3)$. The translation normalization factor

$$D_{\max} = \max_{k,l} \|\mathbf{t}_k - \mathbf{t}_l\|_2 \tag{4}$$

corresponds to the maximum pairwise camera distance within the same rig.

This normalization ensures that all components Δt_{ij}, Δr_{ij}, and Δf_{ij} lie in the range $[0, 1]$, making the metric scale-consistent across translation, rotation, and field-of-view differences. The weighting parameters $\lambda_t, \lambda_r, \lambda_f$ control the relative contribution of the normalized translation, rotation, and FOV terms.

RigV increases with positional spread, rotational disparity, and FOV heterogeneity, yielding a compact descriptor of geometric rig diversity. Higher RigV values indicate rigs with stronger internal variation, which may provide broader scene coverage while simultaneously increasing perception difficulty.

ID	Vehicle	Mounting Positions	Cameras	Description
Controlled Rigs				
R1	SUV	Bumpers, fenders, B-pillars	8×65°	Base configuration
R1-r	SUV	Bumpers, fenders, B-pillars	8×65°	Base with rotation changes
R1-t	SUV	Bumpers, fenders, B-pillars	8×65°	Base with translation changes
R1-f	SUV	Bumpers, fenders, B-pillars	8×90°	Base with FOV changes
R1-c6	SUV	Front bumpers, fenders, B-pillars	6×65°	Base with fewer cameras
R1-c10	SUV	Front bumpers, fenders, B-pillars	10×65°	Base with more cameras
Diverse Rigs				
R2	SUV	Bumpers, B-pillars	4×80°, 2×110°	Large side-camera FOV
R3	SUV	Bumpers, fenders, windshields, mirrors	10×75°	High-density coverage with strong overlap
R4	Sports	Bumpers, fenders, mirrors	8×65°	Sports-car geometry with different mounting semantics
R5	Sports	Mirrors, windshields	4×120°	Few wide-angle cameras
R6	Firetruck	Bumpers, fenders, mirrors, front roof	8×75°, 1×120°	Large vehicle extrema with strong frontal overlap
R7	SUV	Bumpers, front windshield	6×70°, 1×120°	Strong frontal overlap
R8	Sports	Bumpers, mirrors, front windshield	4×65°, 1×100°, 1×120°	Mixed front/back wide FOV
R9	EU HGV	Bumpers, front/back roof, mirrors, fenders	6×75°, 1×120°	Tall-vehicle extrema with strong frontal overlap and roof-mounted front/back cameras

Table 2: Sensor rig configurations used in our benchmark. The *Controlled SUV Family* (R1, R1-r, R1-t, R1-f, R1-c) varies exactly one factor at a time: rotation, translation, FOV, or camera count. This enables a clean validation of our rig-difference metrics. The *Diverse Rigs* (R2–R9) provide realistic variation in camera count, FOV mix, mounting semantics, and vehicle geometry for training and validation.

4.2 Rig Contrastive Distance

Rig Contrastive Distance (RigCD) measures the geometric and optical disparity between two rigs A and B. Unlike RigV, which characterizes the internal heterogeneity of a single configuration, RigCD directly quantifies cross-rig discrepancy and reflects how much a model trained on one rig must adapt to another.

Each camera is represented by its 3D position, full 3D orientation, and FOV. For every camera pair across rigs, we construct the cost matrix

$$C_{ij} = \lambda_t \|\mathbf{t}_i^A - \mathbf{t}_j^B\|_2 + \lambda_r d_R(\mathbf{r}_i^A, \mathbf{r}_j^B) + \lambda_f |f_i^A - f_j^B|,$$

where d_R is the geodesic rotation distance on $SO(3)$. Unlike RigV, this cross-rig matching cost uses the raw translation, rotation, and FOV differences directly. To resolve camera correspondences, we compute the optimal bipartite assignment over the two rigs via minimum-cost matching. Let N_A and N_B denote the number of cameras in rigs A and B, and let $M = \min(N_A, N_B)$. The matched discrepancy is

$$\mathrm{RigCD}_{\mathrm{match}}(A, B) = \frac{1}{M} \sum_{(i,j)\in\mathcal{M}} C_{ij}, \tag{5}$$

where $\mathcal{M}$ is the optimal set of M matched camera pairs. Since rigs may contain different camera counts, we introduce a normalized count-mismatch penalty

$$\mathrm{RigCD}_{\mathrm{count}}(A, B) = \frac{|N_A - N_B|}{\max(N_A, N_B)}, \tag{6}$$

which increases when one rig provides substantially more or fewer viewpoints than the other. The final Rig Contrastive Distance combines these terms:

$$\mathrm{RigCD}(A, B) = \alpha\, \mathrm{RigCD}_{\mathrm{match}}(A, B) + (1 - \alpha)\, \mathrm{RigCD}_{\mathrm{count}}(A, B), \tag{7}$$

with $\alpha \in [0, 1]$ controlling the relative importance of geometric alignment vs. camera-count discrepancy. Componentwise variants RigCD_t, RigCD_r, RigCD_f and $\mathrm{RigCD}_{\mathrm{count}}$ isolate differences in translation, rotation, FOV and count, enabling fine-grained attribution of performance degradation under rig shifts.

5 Evaluation Protocol

In this section we describe the evaluation protocol used in the Plentiful CARLA Camera Rigs benchmark. The protocol measures cross-rig generalization by training perception models on one camera configuration and evaluating them on others. We further outline the baseline models used for evaluation and the calibration procedure for the proposed rig-distance metrics.

5.1 Cross-Rig Evaluation Protocol

The benchmark evaluates how perception models generalize across camera rigs. Given a set of sensor rigs $\mathcal{R} = \{R1, \ldots, R9\}$, a model is trained on the training split of a source rig R_i and evaluated on the test split of a target rig R_j.

This yields a cross-rig transfer matrix $mAP_{\mathrm{rel}}(R_i \to R_j)$ over all source–target rig combinations, from which we compute the relative generalization gap

$$\Delta mAP_{\mathrm{rel}}(R_i \to R_j) = \frac{mAP(R_i \to R_i) - mAP(R_i \to R_j)}{mAP(R_i \to R_i)}. \tag{8}$$

This matrix represents the empirical performance degradation caused by changes in sensor stack configuration.

5.2 Baseline Models

To provide reference performance on the benchmark, we evaluate four representative camera-based multi-view 3D detection architectures: BEVDet [10], BEVFusion [19], Fast-BEV [15], and PETR [21]. BEVDet and BEVFusion lift image features into an explicit BEV representation using LSS-style view transformation, Fast-BEV projects multi-view image features into a 3D voxel representation

for efficient BEV perception, and PETR performs DETR-style object detection by enriching multi-view image features with 3D position embeddings.

All models are trained using their officially released MMDetection3D-based codebases [3]. We adapt the data loaders and nuScenes-style train/evaluation pipelines to the PCCR benchmark format while keeping the original model configurations as close as possible. All experiments use camera input only and predict the nine PCCR benchmark classes. Velocity targets are disabled where supported by the baseline configuration, and velocity is discarded during evaluation. Training is performed with AdamW [24] on two NVIDIA RTX 6000 GPUs with a batch size of four per GPU.

BEVDet We use the camera-only BEVDet configuration with a ResNet-50 [8] image backbone, a CustomFPN image neck, an LSS view transformer, a BEV encoder, and a CenterPoint-style CenterHead detection head [44]. Images are resized to 384×704, and the model is trained for 24 epochs.

BEVFusion For BEVFusion, we use the camera-only configuration. The LiDAR encoder and fusion module are disabled, so the model consists of a ResNet-50 image encoder, an SECONDFPN image neck, an LSS-based view transformation module, a BEV decoder, and a CenterHead detector. Images are resized to 256×704, and the model is trained for 20 epochs.

Fast-BEV For Fast-BEV, we use the multi-frame camera-only configuration with a ResNet-50 backbone, FPN neck, voxel-based multi-view transformation, an M2BEV neck, and an anchor-based FreeAnchor3D detection head. Images are resized to 320×576, and the model is trained for 40 epochs.

PETR For PETR, we use the camera-only PETR3D configuration with GridMask augmentation, a ResNet-50-DCN image backbone, a CPFPN neck, and a PETR transformer decoder. The PETR head uses 900 object queries, 3D sine positional encoding, and multi-view transformer cross-attention. Images are resized to 320×800, and the model is trained for 120 epochs.

5.3 Rig Metric Calibration

The weighting parameters of RigV and RigCD are calibrated using the controlled stacks derived from $R1$. For each variant combination, we use the previously measured $\Delta mAP_{\mathrm{rel}}(R_i \to R_j)$ along with the component-wise distances $(\mathrm{RigCD}_t, \mathrm{RigCD}_r, \mathrm{RigCD}_f, \mathrm{RigCD}_{\mathrm{count}})$. We apply L-BFGS-B optimization and obtain the calibrated parameters $\boldsymbol{\lambda} = \{\lambda_t, \lambda_r, \lambda_f\}$ and α, together with a global scale factor K, yielding the calibrated prediction

$$\widehat{\Delta mAP}_{\mathrm{rel}}(R_i \to R_j) = K \cdot \mathrm{RigCD}(R_i, R_j; \alpha, \boldsymbol{\lambda}). \tag{9}$$

To quantify how well the calibrated metric preserves the ordering of performance degradations under isolated perturbations, we report Spearman rank correlation ρ between $\widehat{\Delta mAP}_{\text{rel}}$ and the observed ΔmAP_{rel}, and compute a 95% confidence interval via bootstrap resampling (sampling stacks with replacement, 5,000 iterations). We evaluate the ranking on the held-out sensor rigs $\{R2, \ldots, R8\}$, which are not used during calibration.

6 Benchmark Results

We present the main findings obtained using the PCCR benchmark. Following the evaluation protocol, we analyze cross-rig transfer behavior and examine how geometric rig differences influence perception performance.

6.1 Cross-Rig Generalization Analysis

Following the cross-rig evaluation protocol described in Section 5, we first report in-domain performance, where each model is trained and evaluated on data from the same rig. Averaged across all rigs, BEVFusion obtains the highest mAP with 0.163 ± 0.052, followed by BEVDet with 0.155 ± 0.053, PETR with 0.149 ± 0.049, and Fast-BEV with 0.090 ± 0.043. These results show that the proposed benchmark supports training and evaluating diverse camera-based 3D detection architectures, even when using compact baseline configurations.

Figure 3 visualizes the resulting cross-rig transfer matrices $\Delta mAP_{\text{rel}}(R_i \rightarrow R_j)$. These results demonstrate that cross-rig domain gaps can be substantial even when the scene content remains identical. Under out-of-distribution (OOD) FOV changes, BEVDet, BEVFusion and PETR experience a near total collapse. We observe similar performance drops using the more complex rigs R6 and R9 with Fast-BEV and PETR. However, training on these higher-variance rigs (R6–R9) improves cross-rig generalization for BEVFusion and BEVDet, leading to smaller relative gaps on unseen rigs. Overall, we observe that cross-rig robustness is highly architecture-dependent.

6.2 Metric Calibration and Evaluation

Using the controlled perturbation rigs, we calibrate RigCD and analyze how geometric factors affect cross-rig transfer in Table 3. Across the reliable baselines, the learned parameters consistently assign the largest weight to camera translation changes, with $\lambda_t = 2.43$ for BEVDet, $\lambda_t = 2.18$ for BEVFusion, and $\lambda_t = 2.28$ for PETR. Rotation receives a moderate weight for BEVFusion and PETR ($\lambda_r = 0.54$ and 0.61), while FOV changes are weighted lower, especially for PETR ($\lambda_f = 0.11$).

RigCD achieves strong ranking agreement on the calibration set for BEVDet, BEVFusion, and PETR, with $\rho = 0.87$, 0.91, and 0.98, respectively. On unseen rig pairs, the correlations remain strong: $\rho = 0.72$ for BEVDet, $\rho = 0.73$ for BEVFusion, and $\rho = 0.80$ for PETR, with 95% CIs of $[0.62, 0.79]$, $[0.65, 0.80]$, and

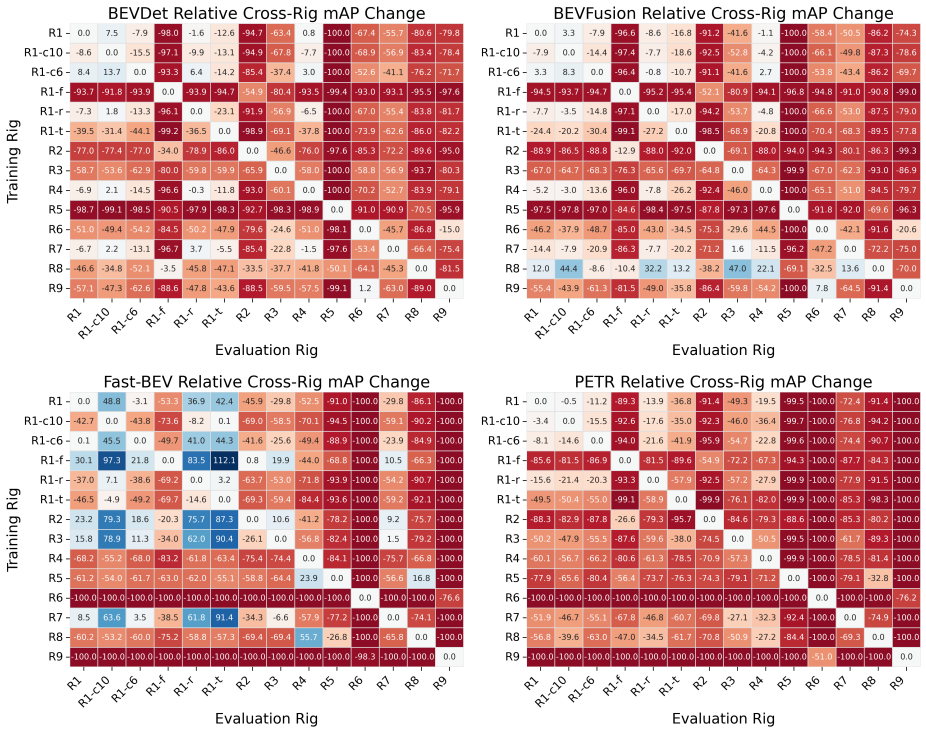

Fig. 3: Relative generalization gaps between rigs, based on mAP. Y-axis shows the rig trained on and X-axis shows the evaluated rigs.

$[0.73, 0.85]$. Fast-BEV is an outlier, with weak calibration and test correlations ($\rho = 0.05$ and 0.10), suggesting that RigCD does not reliably explain its cross-rig behavior under this configuration.

Overall, the calibrated RigCD metric indicates that cross-rig transfer is primarily driven by camera translation differences, followed by rotation, while FOV differences receive lower effective weights for the best-correlated models. While the absolute prediction of mAP remains a challenging and somewhat unstable task due to environmental factors, the consistent ranking performance shown in Figure 4 demonstrates that RigCD provides a dependable proxy for relative rig difficulty.

We show the calibration impact on RigV in Table 4 and observe that cross-rig evaluations on rigs with higher RigV correlate with decreasing mAP.

6.3 Ablation Studies

We conduct ablations to further analyze our benchmark and the metrics.

Component contribution analysis. We perform a component knockout study by vanishing each calibrated weight ($\lambda_t, \lambda_r, \lambda_f$, and α) in turn and re-evaluating the

	Calibration parameters					Controlled Rigs		Diverse R igs	
Model	α	K	λ_t	λ_r	λ_f	ρ	CI	ρ	CI
BEVDet	0.679	0.232	2.427	0.346	0.227	0.866	[0.726, 0.926]	0.718	[0.620, 0.791]
BEVFusion	0.494	0.257	2.178	0.537	0.285	0.914	[0.824, 0.949]	0.734	[0.649, 0.800]
Fast-BEV	0.900	1.000	1.000	1.000	1.000	0.052	[-0.339, 0.423]	0.097	[-0.054, 0.235]
PETR	0.409	0.685	2.284	0.609	0.108	0.982	[0.946, 0.993]	0.804	[0.735, 0.851]

Table 3: RigCD calibration parameters and rank-correlation performance on the calibration (controlled rigs) and test (diverse rigs) splits. We report Spearman's ρ with 95% confidence intervals (CI).

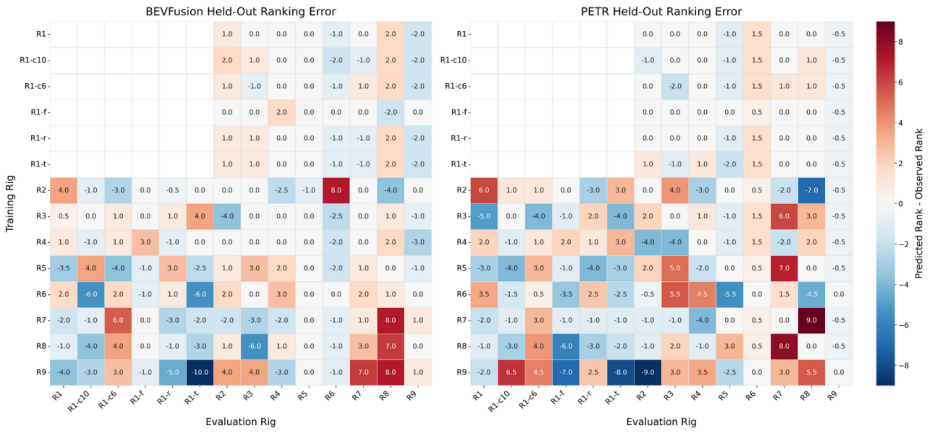

Fig. 4: Signed ranking error heatmap for RigCD-based transfer prediction. Each cell shows the difference between predicted and observed target-rig rank within the test rig set, computed as predicted rank minus observed rank. Values near zero indicate better agreement. Positive and negative values show the direction of rank mismatch.

predictive performance on unseen rigs; the results are shown in Table 5. Across BEVDet and BEVFusion, the RigCD metric relies most strongly on FOV/overlap consistency: removing λ_f reduces Spearman correlation by 0.39 and 0.40, respectively. PETR remains most sensitive to translation errors, with $\lambda_t = 0$ reducing ρ by 0.58. Rotation and camera-count terms have comparatively small effects for most models, suggesting that their calibrated contributions are secondary once translation and overlap structure are accounted for. Fast-BEV exhibits a low full-model correlation and improves when the FOV/overlap term is removed, indicating that its observed degradation pattern is not well explained by the same FOV/overlap weighting that benefits the other BEV-based models. Overall, these findings show that different architectures emphasize different geometric factors, while the integrated RigCD formulation captures the dominant sources of rig-induced domain shift for most models.

Calibration	Controlled Rigs (R1)						Diverse Rigs							
	R1	C10	C6	F	R	T	R2	R3	R4	R5	R6	R7	R8	R9
None	1.09	1.02	1.08	1.09	1.09	1.07	1.30	1.06	1.14	1.33	1.13	1.09	1.29	1.37
BEVDet	1.46	1.36	1.44	1.46	1.46	1.40	1.72	1.44	1.59	1.94	1.50	1.28	1.62	1.98
BEVFusion	1.44	1.34	1.42	1.44	1.44	1.38	1.69	1.42	1.55	1.88	1.47	1.29	1.60	1.92
Fast-BEV	1.09	1.02	1.08	1.09	1.09	1.07	1.30	1.06	1.14	1.33	1.13	1.09	1.29	1.37
PETR	1.53	1.43	1.52	1.53	1.53	1.48	1.78	1.51	1.66	2.00	1.56	1.36	1.68	2.02

Table 4: Rig variance (RigV) values across rigs under different calibration variants.

RigCD Metric Variant	BEVDet		BEVFusion		Fast-BEV		PETR	
	ρ	$\Delta\rho$	ρ	$\Delta\rho$	ρ	$\Delta\rho$	ρ	$\Delta\rho$
Full Model	0.718	0.000	0.734	0.000	0.097	0.000	0.804	0.000
w/o Translation ($\lambda_t = 0$)	0.743	0.025	0.719	-0.015	0.013	-0.084	0.220	-0.584
w/o Rotation ($\lambda_r = 0$)	0.719	0.001	0.738	0.004	0.092	-0.005	0.808	0.003
w/o FOV/Overlap ($\lambda_f = 0$)	0.330	-0.388	0.331	-0.403	0.679	0.582	0.787	-0.018
w/o Camera Count ($\alpha = 1$)	0.716	-0.002	0.733	-0.002	0.097	0.000	0.800	-0.005

Table 5: Component ablation for RigCD. We remove individual metric components and measure the resulting Spearman correlation (ρ) between predicted rig distance and observed relative performance degradation. $\Delta\rho$ denotes the difference.

Multi-model metric calibration. To evaluate the model-agnostic applicability of RigCD, we calibrate and evaluate RigCD as in Section 5.3, but process BEVDet, BEVFusion, Fast-BEV, and PETR jointly instead of separately.

The jointly calibrated RigCD achieves strong generalization on unseen rigs for BEVDet with $\rho = 0.71$ (95% CI: $[0.61, 0.78]$), BEVFusion with $\rho = 0.68$ (95% CI: $[0.58, 0.75]$), Fast-BEV with $\rho = 0.59$ (95% CI: $[0.48, 0.68]$), and PETR with $\rho = 0.74$ (95% CI: $[0.65, 0.80]$). These results indicate that a shared RigCD calibration captures geometric rig differences across architectures, while the remaining variation reflects model-specific sensitivities to particular rig perturbations. Interestingly, this has also significantly improved the results for Fast-BEV, suggesting that models that are harder to calibrate benefit from more data points, even when these are from different models.

Multi-rig training ablation. We study multi-rig training using BEVFusion on a non-overlapping union of rig slices ($R1$, $R2$, $R3$, $R4$, $R5$, $R6$), resulting in a dataset equivalent in size to a single rig. Evaluating on unseen rigs ($R7$, $R8$, $R9$), we measure ΔmAP_{rel} relative to the identity pairs (e.g., $R7 \rightarrow R7$). This yields -38.37% on $R7$, -71.16% on $R8$, and -48.14% on $R9$. While this improves average cross-rig performance and variance, it does not surpass the best single-rig transfer cases.

7 Discussion

General discussion. The benchmark shows that cross-rig generalization is highly architecture-dependent. BEVFusion achieves the best in-domain performance,

followed by BEVDet and PETR, while Fast-BEV performs noticeably worse overall. Under cross-rig evaluation, BEVDet, BEVFusion, and PETR show substantial sensitivity to rig changes, with severe drops under out-of-distribution FOV shifts and on some complex diverse rigs.

RigCD provides a useful ranking proxy for most models. After calibration, it correlates strongly with unseen cross-rig transfer difficulty for BEVDet, BEVFusion, and PETR, reaching $\rho = 0.72$, 0.73, and 0.80, respectively. Fast-BEV is an exception: its weak correlations indicate that its transfer behavior is not well explained by the same calibrated geometric factors.

The calibrated weights show that translation differences are the dominant factor for the reliable baselines, followed by rotation, while FOV receives lower effective weights. The RigV results further indicate that higher geometric rig variance generally corresponds to more difficult transfer, supporting its use as a compact descriptor of rig complexity.

Limitations. The the benchmark intentionally isolates geometric rig variation by rendering identical scenes across rigs. While this enables controlled analysis, it also implies that the benchmark remains simulation-based and therefore subject to the simulation-to-real gap. Our baseline evaluation focuses on two representative models, which limits the generality of architectural conclusions.

RigCD relies solely on calibration metadata and therefore captures only geometric aspects of the observation process. Other sensor characteristics, such as photometric differences, lens distortion, or hardware-specific artifacts, are not modeled. Furthermore, RigCD is defined for pairwise rig comparison and does not directly extend to scenarios where models are trained on multi-rig datasets.

Future work. Future work could extend the benchmark with additional sensor modalities, rig configurations, and environmental diversity. Incorporating real-world datasets would further improve ecological validity. From a modeling perspective, the results motivate approaches explicitly designed for cross-rig robustness, such as rig-aware data augmentation or calibration-conditioned perception models. Another promising direction is uncertainty estimation when deploying perception systems on previously unseen sensor rigs.

8 Conclusion

We introduced the *Plentiful CARLA Camera Rigs* benchmark for studying cross-rig generalization in camera-based automotive perception by rendering identical driving scenes under systematically varied camera configurations. Using this setup, we show that geometric rig variation can induce substantial performance shifts and that cross-rig robustness depends strongly on architectural design. We further propose two calibration-based descriptors, *Rig Variance* and *Rig Contrastive Distance* (RigCD), to quantify rig diversity and cross-rig discrepancy using calibration metadata. Our experiments show that RigCD correlates strongly with observed cross-rig performance degradation and provides a useful proxy for ranking transfer difficulty between sensor rigs.

References

1. Bader, T.A., Eberhardt, T.D., Sohn, T.S., Stork, W., et al.: Toward a universal perception layer: A survey on sensor-agnostic advanced driver assistance systems. Inf. Fusion **136** (2026). https://doi.org/10.1016/j.inffus.2026.104543
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11621–11631 (2020)
3. Contributors, M.: MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d (2020)
4. Dalal, A., Hagen, D., Robbersmyr, K.G., Knausgård, K.M.: Gaussian splatting: 3d reconstruction and novel view synthesis: A review. IEEE Access **12**, 96797–96820 (2024)
5. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proc. Conf. Robot Learn. pp. 1–16 (2017)
6. Embacher, F., Holtz, D., Uhrig, J., Cordts, M., Enzweiler, M.: Neural Rendering for Sensor Adaptation in 3D Object Detection. In: Proc. IEEE Intell. Veh. Symp. pp. 1400–1407 (2025)
7. Gamage, D., et al.: Evaluating sensor configurations for autonomous driving: A perception-entropy-based framework. IEEE Trans. Intell. Veh. (2025)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 770–778 (2016)
9. Hu, C., et al.: Investigating the impact of multi-lidar placement on 3d object detection for autonomous driving. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
10. Huang, J., Huang, G., Zhu, Z., Ye, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
11. Huang, J., Ye, Y., Liang, Z., Shan, Y., Du, D.: Detecting as labeling: Rethinking lidar-camera fusion in 3d object detection. In: Eur. Conf. Comput. Vis. pp. 439–455 (2024)
12. Indu, S., Srivastava, S., Sharma, V.: Optimal camera placement and orientation of a multi-camera system for self driving cars. In: Proc. Int. Conf. Vis., Image Signal Process. pp. 1–5 (2020)
13. Klinghoffer, T., Philion, J., Chen, W., Litany, O., Gojcic, Z., Joo, J., Raskar, R., Fidler, S., Alvarez, J.M.: Towards viewpoint robustness in bird's eye view segmentation. In: Int. Conf. Comput. Vis. pp. 8515–8524 (2023)
14. Li, S., Kachana, P., Chidananda, P., Nair, S., Furukawa, Y., Brown, M.: Rig3r: Rig-aware conditioning and discovery for 3d reconstruction. In: Adv. Neural Inform. Process. Syst. (2025)
15. Li, Y., Huang, B., Chen, Z., Cui, Y., Liang, F., Shen, M., Liu, F., Xie, E., Sheng, L., Ouyang, W., et al.: Fast-bev: A fast and strong bird's-eye view perception baseline. IEEE Trans. Pattern Anal. Mach. Intell. **46**(12), 8665–8679 (2024)
16. Li, Y., Liu, Z.: Information entropy-based viewpoint planning for 3-d object reconstruction. IEEE Trans. Robot. **21**(3), 324–337 (2005)
17. Li, Y., et al.: Influence of camera–lidar configuration on 3d object detection for autonomous driving. In: Proc. IEEE Int. Conf. Robot. Autom. (2024)
18. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. IEEE Trans. Pattern Anal. Mach. Intell. (2024)

19. Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., Tang, Z.: Bevfusion: A simple and robust lidar-camera fusion framework. Adv. Neural Inform. Process. Syst. **35**, 10421–10434 (2022)
20. Liu, M., Yurtsever, E., Fossaert, J., Zhou, X., Zimmer, W., Cui, Y., Zagar, B.L., Knoll, A.C.: A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook. IEEE Trans. Intell. Veh. **9**(11), 7138–7164 (2024)
21. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: Eur. Conf. Comput. Vis. pp. 531–548 (2022)
22. Liu, Y., Yan, J., Jia, F., Li, S., Gao, A., Wang, T., Zhang, X.: Petrv2: A unified framework for 3d perception from multi-camera images. In: Int. Conf. Comput. Vis. pp. 3262–3272 (2023)
23. Liu, Y., et al.: Where should we place lidars on the autonomous vehicle? an optimal design approach. In: Proc. IEEE Int. Conf. Robot. Autom. (2022)
24. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
25. Ma, X., et al.: Perception entropy for evaluating multi-sensor configurations in autonomous driving. In: IEEE Conf. Comput. Vis. Pattern Recog. (2024)
26. Mao, J., Shi, S., Wang, X., Li, H.: 3d object detection for autonomous driving: A comprehensive survey. Int. J. Comput. Vis. **131**(8), 1909–1963 (2023)
27. NVIDIA, Ali, A., Bai, J., Bala, M., Balaji, Y., Blakeman, A., Cai, T., Cao, J., Cao, T., Cha, E., Chao, Y.W., Chattopadhyay, P., Chen, M., Chen, Y., Chen, Y., Cheng, S., Cui, Y., Diamond, J., Ding, Y., Fan, J., Fan, L., Feng, L., Ferroni, F., Fidler, S., Fu, X., Gao, R., Ge, Y., Gu, J., Gupta, A., Gururani, S., El Hanafi, I., Hassani, A., Hao, Z., Huffman, J., Jang, J., Jannaty, P., Kautz, J., Lam, G., Li, X., Li, Z., Liao, M., Lin, C.H., Lin, T.Y., Lin, Y.C., Ling, H., Liu, M.Y., Liu, X., Lu, Y., Luo, A., Ma, Q., Mao, H., Mo, K., Nah, S., Narang, Y., Panaskar, A., Pavao, L., Pham, T., Ramezanali, M., Reda, F., Reed, S., Ren, X., Shao, H., Shen, Y., Shi, S., Song, S., Stefaniak, B., Sun, S., Tang, S., Tasmeen, S., Tchapmi, L., Tseng, W.C., Varghese, J., Wang, A.Z., Wang, H., Wang, H., Wang, H., Wang, T.C., Wei, F., Xu, J., Yang, D., Yang, X., Ye, H., Ye, S., Zeng, X., Zhang, J., Zhang, Q., Zheng, K., Zhu, A., Zhu, Y.: World simulation with video foundation models for physical ai (2025)
28. NVIDIA, Cao, Y., de Lutio, R., Fidler, S., Cobo, G.G., Gojcic, Z., Igl, M., Ivanovic, B., Karkus, P., Esturo, J.M., Pavone, M., Smith, A., Tanimura, E., Tyszkiewicz, M., Watson, M., Wu, Q., Zhang, L.: Alpasim: A modular, lightweight, and data-driven research simulator for autonomous driving (2025), `https://github.com/NVlabs/alpasim`
29. NVIDIA Corporation: Physicalai autonomous vehicles dataset (2025), `https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles`
30. Paulin, G., Ivasic-Kos, M.: Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. Artif. Intell. Rev. **56**(9), 9221–9265 (2023)
31. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Eur. Conf. Comput. Vis. pp. 194–210 (2020)
32. Reichert, H., Lang, L., Rösch, K., Bogdoll, D., Doll, K., Sick, B., Reuss, H.C., Stiller, C., Zöllner, J.M.: Towards Sensor Data Abstraction of Autonomous Vehicle Perception Systems. arXiv preprint arXiv:2105.06896 (2021)
33. Sima, C., Renz, K., Chitta, K., Chen, L., Zhang, H., Xie, C., Beißwenger, J., Luo, P., Geiger, A., Li, H.: Drivelm: Driving with graph visual question answering. In: Eur. Conf. Comput. Vis. pp. 256–274 (2024)

34. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Int. Conf. Comput. Vis. pp. 2446–2454 (2020)
35. Vázquez, P.P., et al.: Viewpoint selection using viewpoint entropy. Comput. Graph. Forum **22**(4), 689–700 (2003)
36. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Automatic view selection using viewpoint entropy and its application to image-based modelling. In: Comput. Graph. Forum. vol. 22, pp. 689–700 (2003)
37. Wakabayashi, K., Yukawa, C., Oda, T., Barolli, L.: A camera placement system for motion analysis and object recognition: System assessment by simulations and an experiment. In: Proc. Int. Conf. Innovative Mobile Internet Services Ubiquitous Comput. pp. 27–38 (2024)
38. Wang, H., Yao, K., Pottie, G., Estrin, D.: Entropy-based sensor selection heuristic for target localization. In: Proc. Int. Symp. Inf. Process. Sensor Netw. pp. 36–45 (2004)
39. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5294–5306 (2025)
40. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Int. Conf. Comput. Vis. pp. 20697–20709 (2024)
41. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Proc. Conf. Robot Learn. pp. 180–191 (2022)
42. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., et al.: Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: Int. Conf. Comput. Vis. pp. 17830–17839 (2023)
43. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 21924–21935 (2025)
44. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. IEEE Conf. Comput. Vis. Pattern Recog. (2021)