

nuCarla: A nuScenes-Style Bird's-Eye View Perception Dataset for CARLA Simulation

Zhijie Qiao¹, Zhong Cao¹, Henry X. Liu^{1,2}

¹ Civil and Environmental Engineering, University of Michigan

² Transportation Research Institute, University of Michigan

<https://github.com/michigan-traffic-lab/nuCarla>

Abstract

End-to-end (E2E) autonomous driving heavily relies on closed-loop simulation, where perception, planning, and control are jointly trained and evaluated in interactive environments. Yet, most existing datasets are collected from the real world under non-interactive conditions, primarily supporting open-loop learning while offering limited value for closed-loop testing. Due to the lack of standardized, large-scale, and thoroughly verified datasets to facilitate learning of meaningful intermediate representations, such as bird's-eye-view (BEV) features, closed-loop E2E models remain far behind even simple rule-based baselines. To address this challenge, we introduce nuCarla, a large-scale, nuScenes-style BEV perception dataset built within the CARLA simulator. nuCarla features (1) full compatibility with the nuScenes format, enabling seamless transfer of real-world perception models; (2) a dataset scale comparable to nuScenes, but with more balanced class distributions; (3) direct usability for closed-loop simulation deployment; and (4) high-performance BEV backbones that achieve state-of-the-art detection results. By providing both data and models as open benchmarks, nuCarla substantially accelerates closed-loop E2E development, paving the way toward reliable and safety-aware research in autonomous driving.

1. Introduction

In the field of autonomous driving, end-to-end (E2E) systems have attracted increasing attention. UniAD [14] represents an influential milestone, proposing a transformer-based architecture that unifies perception, prediction, and planning through a query-driven design, achieving state-of-the-art performance on the large-scale nuScenes [3] dataset. Subsequent works such as VAD [19] and UAD [12] further improved the overall architectural scheme and estab-



Figure 1. Maps of nine CARLA towns with traffic in the nuCarla dataset, shown under diverse weather conditions.

lished new records on open-loop prediction tasks. Despite these advances, studies have highlighted that improvements in open-loop do not necessarily translate to better performance in closed-loop evaluation [2, 44]. In open-loop, the ego agent is reset to the ground truth position at each frame, breaking the causal relationship between actions and outcomes. In contrast, closed-loop evaluation enforces continuous control, where minor disturbances can accumulate and cause the agent to drift off the track [33].

Since real-world testing is risky, inefficient, and often non-reproducible, closed-loop training and evaluation for autonomous vehicles are extensively conducted on simulation environments. Although there have been advances in open-loop E2E modeling, primarily driven by non-interactive real-world datasets [3, 10, 41], these developments have not yet been reflected in the simulation domain. To date, there remains a lack of standardized, thoroughly verified, and large-scale datasets specifically designed for simulation-based closed-loop research.

Among existing platforms, CARLA [9] has become widely used for autonomous driving simulation, offering a high-fidelity, physics simulator with diverse sensor suites and realistic environments. The CARLA autonomous driving leaderboard [4] provides an open framework for evaluating autonomous agents on predefined routes. Despite its popularity, the leaderboard has been criticized for focusing on basic driving skills and lacking rigorous evaluation under complex conditions [5]. To address this, Bench2Drive [17] introduced a large-scale dataset with well-defined metrics and provided pretrained E2E models based on UniAD [14] and VAD [19] architectures. However, the reported driving success rates (SR) for these models remain notably low, at only 16.36% and 15.00%, respectively.

Subsequent works have improved performance on the benchmark. For example, MomAD [35] achieved an SR of 16.71%, VeteranAD [45] 33.85%, DriveTransformer [18] 35.01%, and Orion [11] 54.62%. Nevertheless, even the best-performing framework, Orion, which leverages the advanced Vision-Language Model (VLM) Qwen2 [38] for scene understanding and navigation reasoning, succeeds in only about half of the scenarios, each lasting merely 20 seconds. In contrast, a simple rule-based algorithm, PDM-Lite [34], originally designed for Graph Visual Question Answering tasks, achieves a remarkable 92.27% SR when utilizing ground-truth perception information [1].

We argue that the suboptimal performance of existing models does not necessarily stem from the E2E paradigm itself, but rather from limitations in the available data. *Most simulation datasets provide only raw sensor inputs and direct vehicle control outputs, restricting E2E systems’ ability to learn meaningful intermediate representations, such as bird’s-eye-view (BEV) features, which are critical for improved generalization and stability.* Yet, such intermediate-level datasets and pretrained perception backbones remain largely unexplored in the literature, making it difficult to develop robust closed-loop systems.

To address this gap, we introduce nuCarla, a nuScenes-style, camera-based BEV perception dataset built within the CARLA simulator. Following the standard nuScenes protocol, nuCarla contains 1,000 driving scenarios (700 for training, 150 for validation, and 150 for testing), each consisting of 40 frames sampled at 0.5-second intervals. The dataset strictly aligns with nuScenes in naming conventions, annotation structure, file hierarchy, and API compatibility, enabling direct transfer of existing BEV models to the CARLA environment without modification.

To thoroughly validate the nuCarla dataset, we train and evaluate four state-of-the-art BEV perception models: BEVFormer [24, 42], PETR [26, 27], BEVDet [15, 16], and FastBEV [23]. All models achieve stable convergence and demonstrate strong detection performance on the nuCarla validation set, as measured by the official nuScenes metrics.

We release the full dataset along with pretrained weights for each BEV architecture. In the same way that modern perception frameworks adopt ResNet [13] or VoVNet [20] as standard visual backbones, we envision nuCarla as a BEV-level perception backbone for the development of robust E2E autonomous driving systems.

Finally, we resolve version conflicts by upgrading the legacy mmdetection3d-1.0 [7] frameworks (widely adopted in perception models and E2E systems) to ensure full compatibility with the latest PyTorch and GPU architectures. This resolves a persistent pain point in the research community [31, 32, 37].

The main contributions of this work are as follows:

1. We provide a nuScenes-style BEV perception dataset in the CARLA simulator to facilitate development of perception models.
2. We validate the dataset by training four BEV models, achieving competitive performance under the official nuScenes metrics. We also upgrade the legacy mmdetection3d framework to be compatible with the latest PyTorch and GPU architectures.
3. We release pretrained weights for all evaluated BEV architectures, providing strong perception backbones to support future E2E autonomous driving research.

2. Related Work

2.1. Open-Loop Trajectory Prediction

Early E2E autonomous driving frameworks are primarily implemented and evaluated on open-loop trajectory prediction tasks. UniAD [14] introduces a unified transformer-based architecture that formulates perception, prediction, and planning as interdependent query-based tasks, thereby mitigating error propagation across subtasks and achieving leading results on multiple benchmarks of the nuScenes [3] dataset. VAD [19] proposes a vectorized approach that replaces rasterized inputs with instance-level vector representations, modeling agents and map features as explicit geometric entities for improved interpretability and inference speed. GenAD [46] is a generative framework that produces driving plans from raw sensor inputs by encoding scenes into instance tokens, learning trajectory priors in a latent space, and modeling agent and ego dynamics. Para-Drive [40] explores differentiable modular architectures and designs a fully parallel structure that enhances safety and runtime. Hydra-MDP [25] uses knowledge distillation from both human and rule-based experts and generates diverse trajectory candidates through a multi-head decoder that accounts for multiple evaluation metrics; this method achieved first place in the Navsim challenge [8]. Hydra-MDP++ [21] extends Hydra-MDP [25] by introducing richer behavioral evaluation metrics and a lightweight ResNet-34 backbone, achieving additional gains. UAD [12]

introduces an unsupervised vision-based framework that eliminates manual 3D annotations, using an angular perception pretext to learn spatiotemporal dynamics, achieving greater robustness and efficiency. Despite their architectural differences and methodological advances, these approaches are mainly benchmarked in open-loop settings, breaking causal relationship between decisions and future observations, and may not necessarily lead to good closed-loop driving quality [2, 44].

2.2. Closed-Loop Autonomous Driving

Recognizing the gap between open- and closed-loop evaluations, the research community has increasingly turned to simulation-based platforms to facilitate more comprehensive assessment of E2E systems. Since real-world testing remains expensive and time-consuming for most research groups, CARLA [9] has become widely adopted among existing platforms. For example, VAD-v2 [6] extended its open-loop trajectory prediction tasks by conducting closed-loop evaluations on the Town05 benchmark [4]. However, the code for its closed-loop development is not publicly available, limiting other researchers’ ability to reproduce or build upon these results. UniAD [12] also reported closed-loop testing but does not release its source code.

To address these limitations, Bench2Drive [17] released a large-scale CARLA-based dataset designed to accelerate E2E autonomous driving development, providing pre-trained models based on the UniAD and VAD architectures. However, these models exhibit poor performance, achieving success rates of only 16.36% and 15.00% across 220 evaluation routes, each lasting approximately 20 seconds. Subsequent works report incremental improvements. For example, MomAD (16.71%) [35] introduces momentum to stabilize long-horizon planning by employing topological trajectory matching with Hausdorff distance and cross-attending historical queries. VeteranAD (33.85%) [45] integrates perception directly into planning through a “perception-in-plan” design guided by multi-mode trajectory priors and autoregressive updates. DriveTransformer (35.01%) [18] introduces a task-parallel transformer architecture that enables symmetric interaction among perception, prediction, and planning through sparse queries and streaming updates, improving training stability and scalability. Orion (54.62%) [11] aligns vision-language reasoning with trajectory generation via a unified framework that combines a QT-Former and a generative planner.

However, none of these methods match the performance of the simple rule-based planner PDM-Lite [34], highlighting concerns with current training practices. It also underscores the need to first establish a rigorous perception backbone that effectively captures meaningful intermediate representations, such as bird’s-eye-view (BEV) features, which can then serve as a foundation for further E2E development.

2.3. BEV Perception Models

Bird’s-Eye View (BEV) perception models are broadly categorized into three groups: LiDAR-based, camera-based, and sensor fusion. LiDAR-based approaches such as CenterPoint [43] detects and tracks 3D objects by focusing on their centers using a keypoint detector, while regressing additional attributes such as size, orientation, and velocity. VoxelNet [47] introduces a voxel feature encoding layer that transforms sparse point clouds into unified volumetric features, removing the reliance on manual feature engineering.

Camera-based models are popular due to their lower sensor cost and ability to capture rich semantic information. BEVFormer [24, 42] unifies BEV representations via spatiotemporal transformers, aggregating spatial cues from multiple views and temporal information from past frames. PETR [26, 27] embeds 3D positional information into image features, allowing object queries to access spatial context and enhancing detection accuracy. BEVDet [15, 16] incorporates temporal information by fusing features from adjacent frames, which reduces velocity prediction errors. FastBEV [23] is optimized for efficiency and real-time inference, leveraging lightweight view transformation and multi-frame fusion to deliver both accuracy and speed for on-vehicle deployment.

Sensor fusion approaches combine multiple modalities to improve the accuracy of perception systems. CenterFusion [29] introduces a middle-fusion framework that integrates radar and camera through a frustum-based method. BEVDepth [22] incorporates explicit LiDAR-based depth supervision and a refinement module to address depth estimation challenges in camera-based detection, resulting in robust inference performance.

3. Methodology

In this section, we present the detailed process of constructing the nuCarla dataset, including map selection, weather configuration, traffic generation, ego-vehicle sensor setup, and ground-truth annotation procedures.

3.1. Maps Selection

We construct our dataset from nine distinct CARLA maps, including Town01, Town02, Town03, Town04, Town05, Town06, Town07, Town10, and Mcity Ditigal Twin (a high-fidelity simulation of the real-world autonomous vehicle test facility at the University of Michigan) [28]. Together, these maps represent a rich diversity of driving environments, spanning dense urban centers, suburban neighborhoods, and rural roadways. By leveraging the unique characteristics of each town, such as varying road geometries, intersection layouts, and traffic densities, our dataset provides a comprehensive testbed for evaluating perception models under a wide range of realistic scenarios. Town08

and Town09 maps are not included, as they are reserved for the leaderboard challenge and not publicly available.

In alignment with nuScenes [3], which consists of 1,000 scenarios, our dataset is split into 700 training, 150 validation, and 150 testing scenarios. The 850 training and validation scenarios are evenly distributed across Town01 through Town07, while Town10 and Mcity are reserved for testing in unseen environments.

3.2. Weather Configuration

To enhance diversity, we adopt all 14 predefined weather configurations available in CARLA, encompassing various conditions such as sunny, cloudy, and rainy, as well as different times of day including noon and sunset. For each scenario, a random weather condition is applied from this set, resulting in a final distribution that is approximately uniform across all available weather types.

3.3. Traffic Configuration

The nuCarla dataset includes six object classes, corresponding to the most safety-critical traffic participants: car, truck, bus, pedestrian, motorcycle, and bicycle. The remaining four classes present in nuScenes, namely construction vehicle, trailer, barrier, and traffic cone, are omitted due to practical constraints. Construction vehicles and trailers are not included because CARLA does not provide the necessary object blueprints. For static objects such as barriers and traffic cones, we have not identified a consistent method for placement across diverse environments.

For the included object classes, we incorporated as many variations as possible to enhance visual and behavioral diversity. Specifically, we included 23 types of cars, 3 types of trucks, 3 types of buses, 46 pedestrian models (38 adults and 8 children), 4 types of motorcycles, and 3 types of bicycles. Currently, we include only actively traveling participants, such as moving vehicles, cycles with riders, and walking pedestrians, while excluding stationary ones such as parked vehicles, cycles without riders, or pedestrians sitting or lying down (note that participants temporarily stopped for traffic lights or yielding are treated as traveling). As the primary objective of this project focuses on perception rather than realistic traffic behavior modeling, we did not adopt advanced traffic control workflows, but instead relied on the default CARLA traffic manager to control all participants.

We further adjusted the distribution of generated traffic participants to create a more balanced dataset. In nuScenes, cars and pedestrians dominate the annotations, while motorcycles and bicycles are severely underrepresented. Although this reflects real-world traffic distributions, such imbalance can cause perception models to perform well on frequent classes but underperform on rare ones. For example, the mean Average Precision (mAP) of BEVFormer [24] is 0.618 for cars but only 0.398 for bicycles.

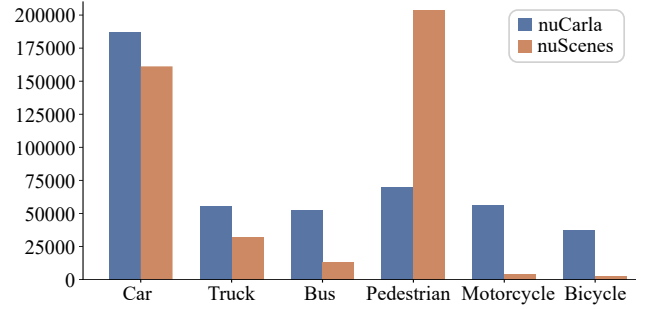


Figure 2. Comparison of class-wise distributions of actively traveling participants in nuCarla and nuScenes across six object classes.

We generated 125 participants for each scenario, including 40 cars, 10 trucks, 10 buses, 25 pedestrians, 20 motorcycles, and 20 bicycles. This ensures sufficient traffic density without causing congestion. Given their larger physical sizes, trucks and buses are more visible than smaller participants such as pedestrians, which compensates for their relatively lower occurrence frequencies. In total, this yields 459,632 annotated samples across six object classes. In comparison, the nuScenes dataset contains 417,609 actively traveling participants across the same six classes, making nuCarla comparable in scale but with a more balanced class distribution. Fig. 2 illustrates the class-wise distributions of actively traveling participants in both datasets.

3.4. Ego and Sensor Configuration

We selected the Nissan Micra ($3.63 \text{ m} \times 1.84 \text{ m} \times 1.50 \text{ m}$) from the available CARLA blueprints to closely match the dimensions of the nuScenes data acquisition vehicle, a Renault Zoe ($4.08 \text{ m} \times 1.78 \text{ m} \times 1.56 \text{ m}$). Six RGB cameras (front, front-left, front-right, back, back-left, and back-right) are mounted on the ego vehicle, with an image resolution of 1600×900 , following the same setup as nuScenes. The intrinsic camera calibration parameters are directly retrieved from CARLA. Note that the dataset does not include any LiDAR or radar sensors. While integrating these sensors is straightforward, this project is primarily focused on camera-based perception. Moreover, introducing insufficiently validated sensor data would risk compromising the reliability of downstream tasks. In future work, we plan to extend the dataset with additional sensing modalities and verify their accuracy through correspondence algorithms [22, 29].

3.5. Ground Truth Annotation

In nuScenes, annotations are created by human experts using specialized labeling tools, which entails significant cost and labor [30]. In CARLA, we can take advantage of privileged access to the ground truth information for all traffic participants, including their size, rotation, and translation.

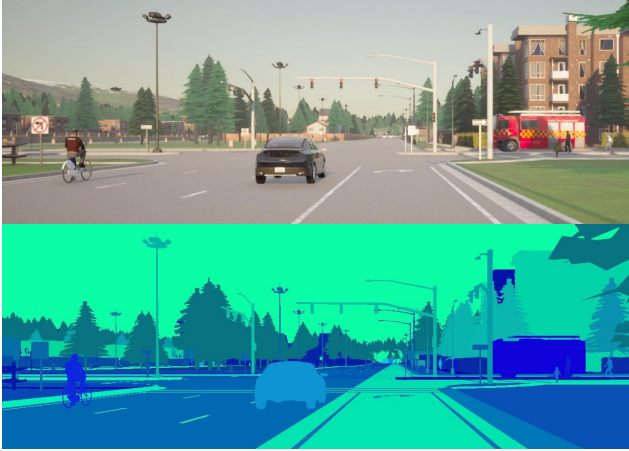


Figure 3. RGB Camera (top) and corresponding instance segmentation camera (bottom) views in CARLA Town06.

However, this also introduces a subtle but important complication: we cannot simply record the ground truth for every participant present in the simulation at a given time. Instead, it is necessary to precisely identify and record only those participants that are visible in at least one of the six camera views. Failing to filter for visibility would lead to significant false positive annotations, which could confuse the model and hinder the learning process.

Since the CARLA RGB camera does not provide information about which traffic participants are captured in its view, we employ an auxiliary instance segmentation camera that assigns a unique pixel value to every object in the scene. For each RGB camera, we mount a corresponding instance segmentation camera at the exact same position with identical calibration, ensuring perfectly aligned views (Fig. 3). This allows us to accurately determine which traffic participants are visible in each camera view and record their corresponding information. Note that the process is used solely for generating ground truth annotations and is not utilized at any stage of model training or inference.

3.6. Data Generation Limitations

We discuss some limitations currently present in the data generation pipeline. In prebuilt CARLA maps, there exist parked vehicles and unattended bicycles or motorcycles that are embedded into the environment as static meshes rather than unique actors. Therefore, they do not appear as distinct instances in the segmentation cameras and lack accessible ground truth, resulting in missing annotations. To resolve this, we edit the CARLA source in Unreal Engine 4 to remove problematic static meshes, then recompile the CARLA package. This produces a custom build that differs from the official release, which may complicate custom data collection for other users.

3.7. nuScenes Formatting

We format our dataset in full alignment with nuScenes, following its naming conventions, annotation format, file hierarchy, and ensuring compatibility with the official Python API. Therefore, nuCarla offers a unique advantage over previous datasets [17, 36, 39]: it provides a standardized closed-loop dataset to which any camera-based BEV perception model originally developed for nuScenes can be transferred without modification. We validate this interoperability by evaluating four state-of-the-art BEV perception models in the subsequent experiment section.

4. Experiment

4.1. Model Configuration

To thoroughly verify nuCarla, including the image quality of the RGB cameras, ground-truth annotation accuracy, and compatibility with nuScenes, we evaluated four state-of-the-art BEV perception models: BEVFormer [24, 42], PETR [26, 27], BEVDet [15, 16], and FastBEV [23]. As our primary objective is to assess their interoperability with our dataset and their overall effectiveness in the closed-loop simulation environment, rather than systematically comparing performance or analyzing their architectural choices, we selected one variant from each model that offers a practical balance between efficiency and performance. Specifically, we used BEVFormer-Base, PETR-VovNet-GridMask-P4-1600x640, BEVDet-R50-4DLongTerm-Stereo-CBGS, and FastBEV-R50-CBGS-4D. For readers unfamiliar with these implementations, we refer to the original publications for further details on network implementation, training schedules, and resource utilization.

4.2. MMDetection3D Upgrade

We address practical compatibility concerns associated with existing E2E models [31, 32, 37], which are built upon the MMDetection3D-1.0 [7] framework. This framework supports only earlier versions of PyTorch (<2.0) and CUDA (<12.0), limiting deployment on newer hardware such as NVIDIA H100 and GeForce RTX 50 series. While the upgraded MMDetection3D-2.0 framework has been released, migrating existing models would require extensive and potentially risky code refactoring.

To address this issue, we implemented a series of targeted patches to upgrade the MMDetection3D-1.0 framework for compatibility with the latest versions of PyTorch (2.7), CUDA (12.8), and modern GPUs. Our approach preserves the original model codebase, introducing only minimal changes necessary to resolve version conflicts. We validated the effectiveness of these modifications by successfully running the selected BEV models on both the official nuScenes dataset and our newly developed nuCarla dataset.

Table 1. Summary of bird’s-eye-view (BEV) perception results for four models on the nuCarla *validation* set, evaluated on seven training maps and six object classes using the official nuScenes detection metrics.

Methods	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
BEVFormer (Base) [24, 42]	0.813	0.266	0.063	0.053	0.728	0.177	0.778
PETR (VovNet-GridMask-P4-1600x640) [26, 27]	0.745	0.438	0.088	0.061	0.906	0.137	0.710
BEVDet (R50-4DLongTerm-Stereo-CBGS) [15, 16]	0.811	0.272	0.082	0.117	0.858	0.197	0.753
FastBEV (R50-CBGS-4D) [23]	0.777	0.302	0.093	0.141	0.875	0.193	0.728

Table 2. Summary of bird’s-eye-view (BEV) perception results for four models on the nuCarla *test* set, evaluated on the two unseen test maps and six object classes using the official nuScenes detection metrics.

Methods	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
BEVFormer (Base) [24, 42]	0.579	0.451	0.068	0.173	1.006	0.208	0.599
PETR (VovNet-GridMask-P4-1600x640) [26, 27]	0.514	0.596	0.098	0.144	0.852	0.186	0.569
BEVDet (R50-4DLongTerm-Stereo-CBGS) [15, 16]	0.560	0.452	0.093	0.533	1.245	0.265	0.546
FastBEV (R50-CBGS-4D) [23]	0.509	0.489	0.115	0.686	1.474	0.274	0.498

4.3. Training and Evaluation Settings

All models were trained from scratch on H100 GPUs for 24 epochs using the 700 training scenarios. Performance was evaluated on the 150 validation scenarios from the 7 trained maps, as well as the 150 test scenarios from the 2 unseen maps. The results were reported following the official nuScenes detection metrics over the six available classes, including average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE), and average attribute error (AAE). Mean Average Precision (mAP) and nuScenes Detection Score (NDS) were used to summarize overall performance.

4.4. Validation Results

Evaluation results of the four BEV models on the nuCarla validation set are shown in Table 1. Overall, BEVFormer achieves the best results. Nevertheless, all models demonstrate robust performance, with mAP and NDS consistently exceeding 0.7. Note that since only a single variant from each model family was selected, these results are intended to demonstrate practical viability rather than provide a comprehensive benchmark.

Compared to their validation results on nuScenes, all models achieve substantially higher mAP and NDS on nuCarla, which we attribute to two primary factors. First, nuCarla excludes trailers and construction vehicles, which are infrequently annotated in nuScenes and exhibit relatively low performance (e.g. BEVFormer reports only 0.172 mAP for trailers and 0.129 mAP for construction vehicles). Their exclusion prevents these categories from adversely affecting the overall scores. Second, the reduced number of classes simplifies the problem, potentially lowering training complexity and facilitating rapid convergence.

4.5. Test Results

On the nuCarla test set (Table 2), all models generalize reasonably well, despite performing worse than on the validation set. This is because the test set includes two previously unseen and practically more challenging maps. For example, Town10 is the flagship map of CARLA, featuring dense, vivid urban environments, whereas Mcity is a large open area with many traffic participants simultaneously in view. In contrast, on nuScenes, model performance on the validation and test sets does not differ significantly, suggesting that the distributions are similar. Overall, these results underscore the importance of evaluating model generalization on diverse and challenging scenarios.

4.6. Visualization

Fig. 4 shows a visual illustration of model predictions compared to ground truth. The predictions are generated by BEVFormer, evaluated on a sample from the Town03 map. To improve clarity, we display only the three front-facing cameras, rather than all six available views. As shown, the model predictions closely match the ground truth in translation, rotation, and size, accurately capturing all actors as indicated by the bounding boxes. The only missed prediction is a firetruck that is partially visible behind a statue on the far side of the roundabout. However, this object is also barely visible to humans due to backlighting, and its omission does not present any immediate safety concern.

4.7. Per-Class Metric Comparison

To better assess the strengths and limitations of the dataset, Table 3 provides a detailed per-class comparison between nuCarla (shown on the left side of each cell) and nuScenes (on the right side), using BEVFormer on the validation set.

Table 3. Per-class detection metrics across six object classes, evaluated by BEVFormer, comparing nuCarla (on the left side of each cell) and nuScenes (on the right side) *validation* set.

Class	AP \uparrow	ATE \downarrow	ASE \downarrow	AOE \downarrow	AVE \downarrow	AAE \downarrow
Car	0.792 0.618	0.269 0.462	0.098 0.152	0.027 0.067	0.896 0.325	0.289 0.196
Truck	0.828 0.370	0.264 0.726	0.042 0.212	0.027 0.093	0.691 0.348	0.251 0.193
Bus	0.826 0.444	0.251 0.753	0.035 0.212	0.039 0.099	0.918 0.868	0.110 0.270
Pedestrian	0.714 0.494	0.359 0.642	0.162 0.295	0.157 0.433	0.404 0.360	0.001 0.175
Motorcycle	0.849 0.429	0.242 0.639	0.028 0.257	0.032 0.438	0.914 0.530	0.195 0.265
Bicycle	0.867 0.398	0.208 0.554	0.011 0.272	0.037 0.484	0.547 0.248	0.217 0.024

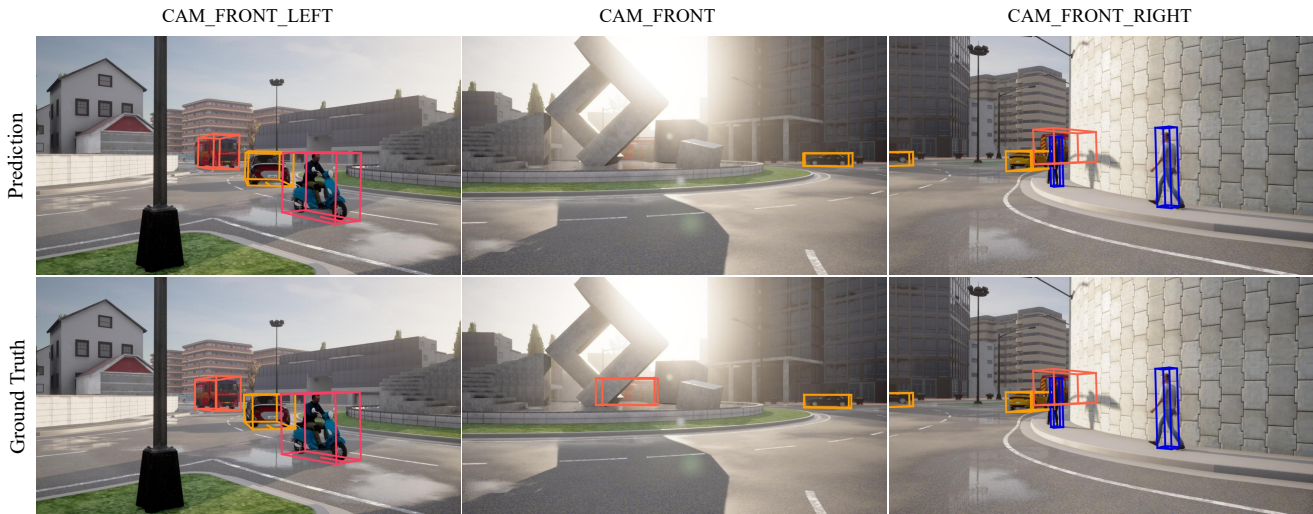


Figure 4. BEVFormer model predictions (top row) and ground truth annotations (bottom row) on a Town03 sample, featuring the front, front-left, and front-right camera views. 3D bounding boxes are colored according to the nuScenes colormap: cars (amber orange), trucks (coral red), pedestrians (blue), and cyclists (pink-red).

In nuScenes, the car category achieves notably higher Average Precision (AP) than the others, primarily due to the dominance of car annotations. In contrast, nuCarla exhibits more uniform scores, reflecting its balanced distribution.

For the Average Translation Error (ATE), Average Scale Error (ASE), and Average Orientation Error (AOE), varying degrees of improvement are also observed in nuCarla, particularly for underrepresented classes. While part of this improvement can be attributed to the more balanced class distribution, we believe another contributing factor is the inherently simpler and more structured nature of the CARLA environment. In this setting, traffic participants follow well-defined and consistent motion patterns, allowing models to effectively learn and generalize spatial relationships.

Conversely, the Average Velocity Error (AVE) is much higher in nuCarla. In nuScenes, the substantial proportion of stationary participants yields zero-velocity samples that are trivial to predict, thereby lowering the overall error. In contrast, all actors in nuCarla are actively traveling, which makes velocity estimation more challenging.

Finally, the Average Attribute Error (AAE) does not display a consistent pattern of discrepancies between the two datasets. This suggests that certain object classes within each dataset may be inherently easier to learn than others, such as pedestrians in nuCarla and bicycles in nuScenes. Additionally, some models demonstrate stronger attribute prediction capabilities; for example, PETR achieves better results in attribute estimation, even though its performance on other metrics is comparatively weaker.

5. Conclusion

In this work, we present nuCarla, a nuScenes-style bird’s-eye-view perception dataset designed for the CARLA simulator. To thoroughly validate nuCarla, we evaluate four state-of-the-art BEV perception models and provide pre-trained weights. This facilitates the learning of meaningful intermediate representations and supports the advancement of end-to-end autonomous driving research through comprehensive closed-loop testing.

References

- [1] autonomousvision. carla_garage: [iccv'23] hidden biases of end-to-end driving models and a starter kit for the carla leaderboard 2.0. https://github.com/autonomousvision/carla_garage, 2023. GitHub repository. 2
- [2] Mohamed-Khalil Bouzidi, Christian Schlauch, Nicole Scheuerer, Yue Yao, Nadja Klein, Daniel Göhring, and Jörg Reichardt. Closing the loop: Motion prediction models beyond open-loop benchmarks. *arXiv preprint arXiv:2505.05638*, 2025. 1, 3
- [3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *CVPR*, 2020. 1, 2, 4
- [4] CARLA. Carla autonomous driving leaderboard. <https://leaderboard.carla.org/>, 2025. 2, 3
- [5] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231, 2022. 2
- [6] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024. 3
- [7] MMDetection3D Contributors. MMDetection3D: Open-MMLab next-generation platform for general 3D object detection. <https://github.com/open-mmlab/mmdetection3d>, 2020. 2, 5
- [8] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2
- [9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16. PMLR, 2017. 2, 3
- [10] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9710–9719, 2021. 1
- [11] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkan Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. *arXiv preprint arXiv:2503.19755*, 2025. 2, 3
- [12] Mingzhe Guo, Zhipeng Zhang, Yuan He, Ke Wang, Liping Jing, and Haibin Ling. End-to-end autonomous driving without costly modularization and 3d manual annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 1, 2, 3
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2
- [14] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023. 1, 2
- [15] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 2, 3, 5, 6
- [16] Junjie Huang, Guan Huang, Zheng Zhu, Ye Yun, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2, 3, 5, 6
- [17] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS 2024 Datasets and Benchmarks Track*, 2024. 2, 3, 5
- [18] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. *arXiv preprint arXiv:2503.07656*, 2025. 2, 3
- [19] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. *ICCV*, 2023. 1, 2
- [20] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 2
- [21] Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025. 2
- [22] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI conference on artificial intelligence*, pages 1477–1485, 2023. 3, 4
- [23] Yangguang Li, Bin Huang, Zeren Chen, Yufeng Cui, Feng Liang, Mingzhu Shen, Fenggang Liu, Enze Xie, Lu Sheng, Wanli Ouyang, et al. Fast-bev: A fast and strong bird's-eye view perception baseline. *arXiv preprint arXiv:2301.12511*, 2023. 2, 3, 5, 6
- [24] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 2, 3, 4, 5, 6

- [25] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024. 2
- [26] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 2, 3, 5, 6
- [27] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 2, 3, 5, 6
- [28] Mcity. Mcity digital twin. <https://github.com/mcity/mcity-digital-twin>, 2025. 3
- [29] Ramin Nabati and Hairong Qi. Centerfusion: Center-based radar and camera fusion for 3d object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1527–1536, 2021. 3, 4
- [30] nuTonomy. *nuScenes DevKit*. GitHub, 2025. Accessed: 2025-10-29. 4
- [31] OpenDriveLab. Available at: <https://github.com/OpenDriveLab/UniAD/issues/206>, 2024. GitHub issue. 2, 5
- [32] OpenDriveLab. Available at: <https://github.com/OpenDriveLab/UniAD/issues/245>, 2025. GitHub issue. 2, 5
- [33] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 1
- [34] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer, 2024. 2, 3
- [35] Ziyang Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don’t shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22432–22441, 2025. 2, 3
- [36] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21371–21382, 2022. 5
- [37] Fundamental Vision. Available at: <https://github.com/fundamentalvision/BEVFormer/issues/313>, 2025. GitHub issue. 2, 5
- [38] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2
- [39] Tianqi Wang, Sukmin Kim, Ji Wenxuan, Enze Xie, Chongjian Ge, Junsong Chen, Zhenguo Li, and Ping Luo. Deepaccident: a motion and accident prediction benchmark for v2x autonomous driving. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2024. 5
- [40] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024. 2
- [41] Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, Ben Sapp, Mingxing Tan, Jyh-Jing Hwang, and Dragomir Anguelov. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios, 2025. 1
- [42] Chenyu Yang, Yuntao Chen, Haofei Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Y. Qiao, Lewei Lu, Jie Zhou, and Jifeng Dai. Bevformer v2: Adapting modern image backbones to bird’s-eye-view recognition via perspective supervision. *ArXiv*, 2022. 2, 3, 5, 6
- [43] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 3
- [44] Jiang-Tian Zhai, Ze Feng, Jihao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscen. *arXiv preprint arXiv:2305.10430*, 2023. 1, 3
- [45] Bozhou Zhang, Jingyu Li, Nan Song, and Li Zhang. Perception in plan: Coupled perception and planning for end-to-end autonomous driving, 2025. 2, 3
- [46] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 87–104. Springer, 2024. 2
- [47] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 3