Preliminary Investigation into Data Scaling Laws for Imitation Learning-Based End-to-End Autonomous Driving

Yupeng Zheng[1,2], Zhongpu Xia[2]*, Qichao Zhang[1]†, Teng Zhang[2], Ben Lu[2], Xiaochuang Huo[2], Chao Han[2], Yixian Li[2], Mengjie Yu[2], Bu Jin[2], Pengxuan Yang[1], Yuhang Zheng[1], Haifeng Yuan[2], Ke Jiang[2], Peng Jia[2], Xianpeng Lang[2], and Dongbin Zhao[1]
[1]Institute of Automation, Chinese Academy of Sciences, [2]Li Auto,

Abstract

The end-to-end autonomous driving paradigm has recently attracted lots of attention due to its scalability. However, existing methods are constrained by the limited scale of real-world data, which hinders a comprehensive exploration of the scaling laws associated with end-to-end autonomous driving. To address this issue, we collected substantial data from various driving scenarios and behaviors and conducted an extensive study on the scaling laws of existing imitation learning-based end-to-end autonomous driving paradigms. Specifically, approximately 4 million demonstrations from 23 different scenario types were gathered, amounting to over 30,000 hours of driving demonstrations. We performed open-loop evaluations and closed-loop simulation evaluations in 1,400 diverse driving demonstrations (1,300 for open-loop and 100 for closed-loop) under stringent assessment conditions. Through experimental analysis, we discovered that (1) the performance of the driving model exhibits a power-law relationship with the amount of training data; (2) a small increase in the quantity of long-tailed data can significantly improve the performance for the corresponding scenarios; (3) appropriate scaling of data enables the model to achieve combinatorial generalization in novel scenes and actions. Our results highlight the critical role of data scaling in improving the generalizability of models across diverse autonomous driving scenarios, assuring safe deployment in the real world. Project repository: https://github.com/ucaszyp/Driving-Scaling-Law

1. Introduction

End-to-end autonomous driving has gained significant attention in recent years. It typically employs a differentiable model that takes raw sensor data as input and generates a potential planning trajectory as output. This paradigm allows for the direct optimization of the entire system in a data-driven manner, offering scalability where performance improvements can be achieved by increasing training data, as exemplified by scaling laws[11, 16, 31].

Some previous works [2, 19] have tried to investigate the effect of scaling up. However, they share the same challenge: insufficient real-world data, resulting in data scaling laws in end-to-end autonomous driving remaining under-explored. As shown in Table. 1, the open-source datasets [4, 5] are typically thousand-scale, far less than the million-scale or billion-scale vision-language data in language models or generative models. Although simulators like CARLA [9] offer a promising solution to the data lack, the huge domain gap hinders its real-world application. Consequently, it remains under-explored whether end-to-end autonomous driving has such common data scaling laws and how can autonomous driving vehicles benefit from the laws.

In this paper, we delve into the data scaling laws of end-to-end autonomous driving in the real world. We aim to investigate the three critical questions:

- *Is there a data scaling law in the field of end-to-end autonomous driving?*
- *How does data quantity influence model performance when scaling training data?*
- *Can data scaling endow autonomous driving cars with the generalization to new scenarios and actions?*

To answer the questions, we collect and annotate a million-scale dataset named ONE-Drive, which contains over 4 million driving demonstrations (about 30,000 hours) of real-world data in diverse cities and road conditions as shown in Figure. 1. Based on the dataset, we first conduct a detailed analysis of the relationship between training data and model performance. Further, we split the scenarios into 23 types based on the traffic condition and agent behavior to delve into the generalization of the data scaling law. Lastly, we set the training data with different distributions on scenario types to analyze the relationship between data scaling and data distribution. For a comprehensive analysis, we

*Project leader
†Corresponding author

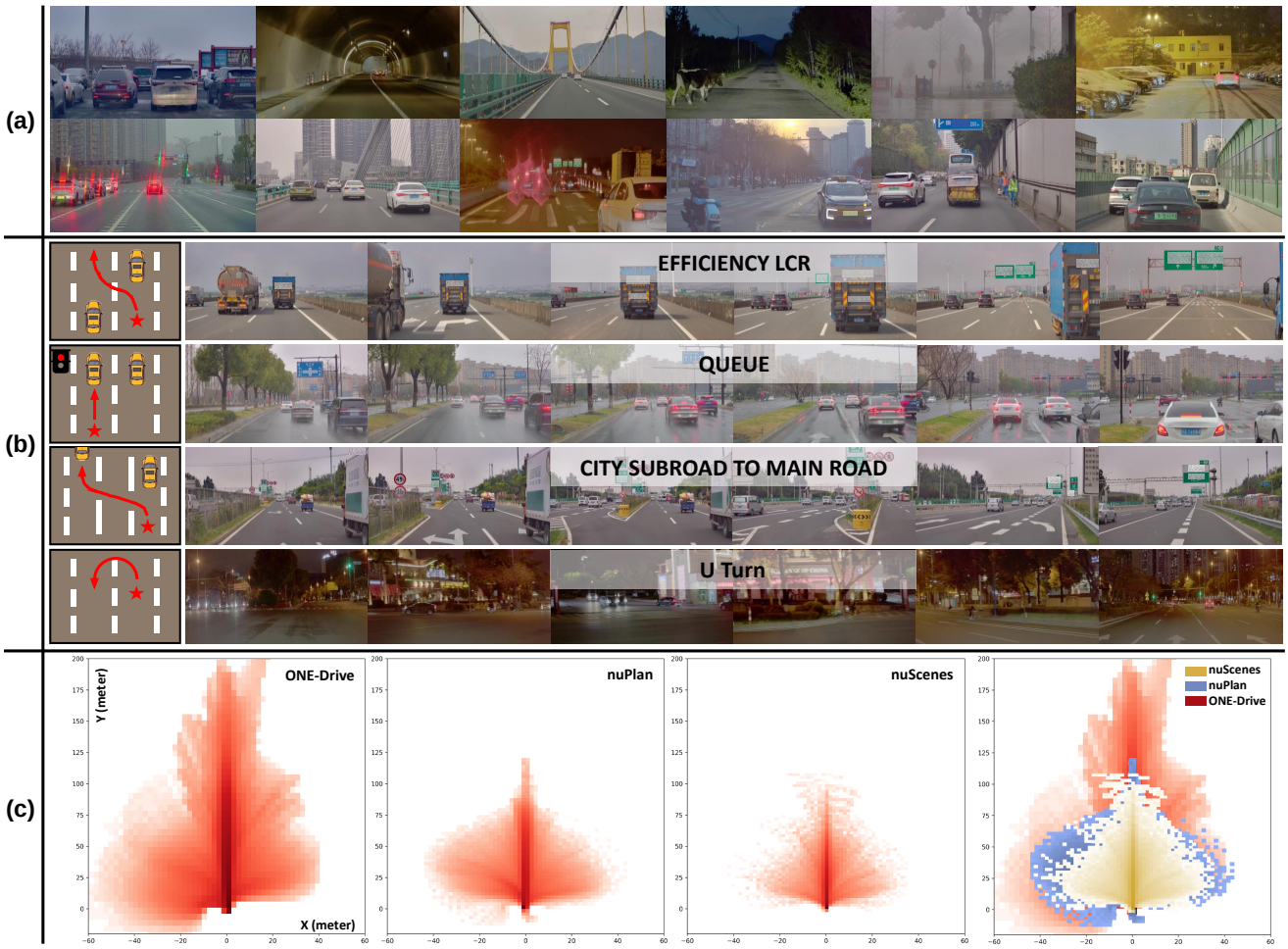

Figure 1. We have collected and utilized a large-scale, diverse real-world dataset, enabling us to investigate data scaling laws of end-to-end driving. Figure (a) illustrates the diversity of our dataset, encompassing various weather conditions, road types, and traffic scenarios. Figure (b) presents 23 scenario types that we have identified to conduct in-depth analyses of the impact of data scale on generalization and the importance of data distribution. Figure (c) compares the trajectory distributions of existing datasets nuScenes[4] and nuPlan[5] with ours. Our trajectory distribution exhibits greater diversity, including a higher proportion of high-speed driving, turning, and lane changes.

conduct the open-loop test, close-loop test, and real-world deployment to evaluate the planning results.

Throughout our experiments, we surprisingly find:

- **Power-laws.** Trajectory fitting capability of end-to-end autonomous driving exhibits a power-law relationship with the number of training data (Section 5.1).
- **Importance of increasing the quantity of targeted data.** Targeted augmentation of long-tailed scenarios exhibits superior efficiency compared to indiscriminate expansion of the overall dataset. (Section 5.2).
- **Combinatorial generalization.** With the scaling of training data, models gradually acquire combinatorial generalization capabilities, enabling them to combine known information to achieve generalizable planning for new scenarios and actions (Section 5.3).

Table 1. Comparison among the previous dataset and our introduced ONE-Drive for the end-to-end autonomous driving task.

Dataset	Setting		Source	Scale
	Open-loop	Visual closed-loop		(demonstrations)
Bench2drive [14]		✓	Simulator	10k
nuScenes [4]	✓		Real-world	1k
nuPlan [5]	✓		Real-world	20k
ONE-Drive	✓	✓	Real-world	**4M**

2. Related Work

2.1. End-to-end Autonomous Driving

Recent years have witnessed the development of autonomous driving vehicles [2, 7, 13, 15, 19, 29, 30]. Among them, end-to-end approaches catch great attention for their direct optimization of the whole system, offering a poten-

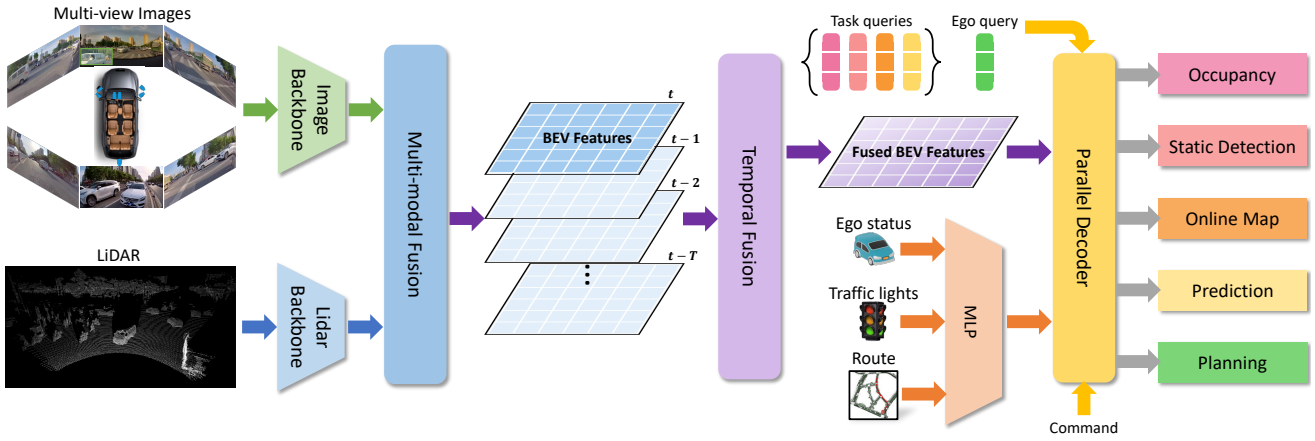

Figure 2. To better investigate the data scaling laws of end-to-end algorithms, we implement a parallel baseline methodology. This approach, inspired by the PARA-Drive [33] framework, offers enhanced training stability and computational efficiency.

tial solution for the information loss and cascade error of traditional module-based approaches. They typically take raw sensor data as input and generate a potential path or plan as output. Some previous works focus on end-to-end solutions in simulators like carla [9]. For example, TransFuser [7] and InterFuser [29] combine information from different sensors to enhance the robustness and performance of the autonomous driving model in complex scenarios by fusing perception and planning. TCP [34] demonstrates exceptional performance using only a monocular camera by introducing target trajectory-guided control prediction, surpassing other methods that use multi-sensor inputs (multi-camera and LiDAR). AD-MLP [36] presents a simple method based on a multilayer perceptron (MLP) that directly outputs the future trajectory of the vehicle from raw sensor data input. However, the domain gap between the simulator and the real world hinders their industrial practice. To avoid the domain gap, some work [13] tries end-to-end autonomous planning in the real world. For example, UniAD [13] utilizes BEV queries to unify different tasks with the transformer, and VAD [15] proposes to represent the scene in a vectorized space. More recently, BEV-Planner [20] proposes a novel metric that enables a more comprehensive evaluation of model performance. Such data-driven optimization enables the ability to improve the system by simply scaling training resources. The end-to-end approaches offer a potential solution for the information loss and cascade error of traditional module-based approaches.

2.2. Scaling Laws

In recent years, the development of foundation models has highlighted the significance of scaling laws in the field of large language models[11, 16], vision generative models[1, 8, 11, 23, 26, 37], and robotics[3, 18, 22, 25, 28, 32, 38]. These laws elucidate the relationship between dataset size, model size, and performance, showing that the effectiveness of transformer-based models follows a power-law function relative to the number of model parameters, the volume of training data, and the computational resources used for training. This concept has also been extended to other domains, such as image and video synthesis[31], allowing for a more efficient trade-off between resource allocation and performance outcomes. However, there is limited research on the application of scaling laws to end-to-end autonomous driving, primarily due to constraints related to data availability. It still remains uncertain whether any scaling law exists that links the scalability of neural networks to planning results in autonomous driving, which is the main focus of our work.

3. Method

Following PARA-Drive [33], we implement a parallel modular end-to-end autonomous driving algorithm named ONE model, as shown in Figure. 2. The input is LiDAR point clouds and panoramic RGB images and the expected output is a planning trajectory.

3.1. BEV Encoder

Given the input of multi-view images and point clouds, we generate a unified Bird's Eye View (BEV) representation through the View Transformation module and the Temporal Fusion module.

View Transformation. For multi-view camera images, we utilize an image backbone to extract image features. Then we utilize LSS [27] to project these image features into a 3D frustum form. For LiDAR point clouds, we extract voxelized LiDAR features with 3D sparse convolution layers. The features are then flattened along the height dimension, leading to features in a BEV plane. Finally, the image

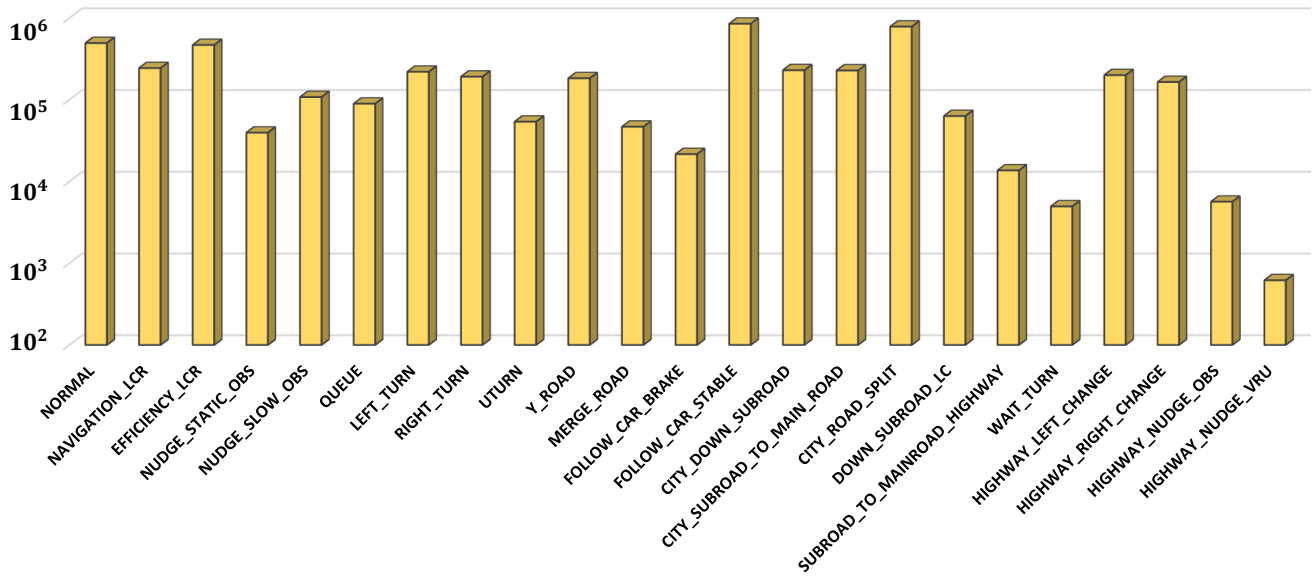

Figure 3. Distribution of the 23 scenario types.

features and point cloud features are fused with squeeze-and-excitation (SE) blocks [12].

Temporal Fusion. To effectively leverage historical information, we employ a temporal fusion module. We set up a FIFO queue to sequentially extract and store past BEV features and relative poses. During training and inference, the features stored in the queue are transformed using the pose information to align with the current pose. The transformed features are then fused with the current BEV features by SE blocks [12]. Finally, the queue is updated by adding the current frame features to the end of the queue and dequeuing the front element of the queue.

3.2. Parallel Decoder

Our decoder comprises **five tasks**: online map prediction, 4D occupancy prediction, static detection, motion prediction, and planning. Inspired by [13, 15, 30], we design each module with learnable query features tailored specifically for its corresponding task.

Online Mapping. For online map prediction, we utilize a set of map queries to estimate a vector map from BEV features, along with the class scores corresponding to each map vector. The road map labels include lane dividers, lane centerlines, crosswalks, and stop lines. We use the loss function defined in MapTR [21].

4D Occupancy Prediction. For occupancy, we establish a set of occupancy queries to estimate the 3D occupancy and the corresponding flow for the next 3 seconds from BEV features. The 3D occupancy prediction loss is the same as [6] and the flow loss employed in Cam4DOcc [24].

Static Detection. We specifically design an additional static object detection task to address the challenge of capturing small static objects in occupancy prediction. We employ Hungarian matching to match the query features with ground truth values.

Motion Prediction. For dynamic objects, we follow [15] by setting a set of agent queries. Each agent query interacts with the BEV features through attention layers to capture environmental information. Similarly, we supervise the motion prediction of dynamic objects using the L1 loss.

Ego Motion Planning. In the planning module, to fulfill the requirements of real-world planning for precise adherence to navigation, we employ a multi-layer perceptron (MLP) to encode the traffic light status, road-level route (represented by a collection of dense waypoints at the road level), and the motion state of the ego-vehicle, which are served as contextual information for planning. Following [15, 33], we leverage learnable embeddings to construct ego-vehicle queries. These queries are passed as input to the cross-attention layers to interact with BEV features. Then the queries are used to predict multi-modal future planning trajectories with corresponding scores. During inference, we execute the trajectory with the highest confidence score.

3.3. Loss

The overall loss function is defined as the sum of the task losses:

$$\mathcal{L} = \mathcal{L}_{depth} + \mathcal{L}_{map} + \mathcal{L}_{occ} + \mathcal{L}_{static} + \mathcal{L}_{mot} + \mathcal{L}_{plan} \quad (1)$$

where the L_{depth} represents an auxiliary depth loss of forward projection in LSS [27].

Table 2. Training steps and resources of different training dataset sizes (demonstrations).

Dataset Size (Demonstrations)	Training Resource	Training Steps
10 thousand	64 A100	6.0×10^5
50 thousand	64 A100	8.6×10^5
0.7 million	320 A100	9.3×10^6
2 million	512 A100	2.2×10^7
4 million	512 A100	3.1×10^7

4. Experiments Setup

4.1. Implementation Details

Data Collection. The scene images were captured horizontally by seven cameras covering a 360° field of view (FOV), with each image having a resolution of 3840×2160. Additionally, we employed a 128-beam LiDAR sensor to acquire point cloud of the environment.

Baseline. Our method predicts a 6-second future trajectory with 2-second history information. It uses ResNet34 [10] as the default image backbone. The image resolution is downsampled to 512×960 for training. The default BEV resolution is 232×80 for a perception range of $139.2m \times 48m$ longitudinally and laterally. The default hidden state and embedding dimension are 256. We construct five training datasets with varying numbers of demonstrations: **10 thousand**, **50 thousand**, **0.7 million**, **2 million**, and **4 million**. We train each model to full convergence during training for a fair comparison. This allows models trained on different dataset sizes to undergo different numbers of training steps. The specific training steps and computational resources are shown in Table. 2

4.2. Data Mining

To facilitate more flexible adjustment of data distribution, we define 23 scenario types according to the agent behavior and traffic conditions. They can be acquired based on the following meta-information:

- **Navigation Information**: This includes upcoming road names, road types, number of lanes, and distance to intersections.
- **Static Perception**: This includes lane markings, driveable areas, intersection areas, and pedestrian walkways.
- **Dynamic Perception**: This covers road obstacle information, including position, velocity, dimensions, historical trajectories, and predicted future trajectories.
- **Ego Vehicle Status**: This comprises the ego vehicle's position, velocity, acceleration, historical trajectory, and future trajectory.

The distribution of these scenario types in the training dataset (4 million demonstrations) is shown in Figure. 3. For more details about the specific characteristics of the 23 scenario types, please see the supplementary materials.

4.3. Evaluation

To comprehensively analyze the scaling laws of data, we evaluate our model using three methodologies: open-loop evaluation and closed-loop simulation. It is important to acknowledge that, owing to constraints in the reconstruction process, the closed-loop evaluation environment has not been fully aligned with the open-loop. Future work will align the two evaluation settings to make a fair comparison and analysis of the data scaling law in these two settings.

Open-loop Evaluation. The open-loop evaluation computes the Average Displacement Error (ADE) metric between the predicted trajectories and the ground truth trajectories, which can evaluate the network's ability to fit the ground truth trajectory.

Closed-loop Simulation. For closed-loop simulation, we employ 3D-GS [17] to reconstruct partial test scenarios, enabling end-to-end simulation with visual closed-loop feedback. This simulation evaluates the driving trajectory in five dimensions: **safety**, **comfort**, **rule**, **efficiency**, and **navigation**. For each metric, a higher score indicates better performance. Finally, the total score is weighted by these five scores.

$$\begin{aligned} score =& 0.25 \times \text{safety} + 0.15 \times \text{comfort} + 0.2 \times \text{rule} \\ &+ 0.25 \times \text{efficiency} + 0.15 \times \text{navigation} \end{aligned}$$

The implementation details of the closed-loop simulation, along with the evaluation metrics and visualization results, are presented in the supplementary materials.

5. Experiments

In this section, we investigate the relationship between the scale of training data and the ability of end-to-end driving models to fit expert trajectories (Section 5.1.1). Subsequently, we discover the inconsistency in data scaling laws between open-loop and closed-loop evaluations, revealing that merely increasing data volume does not indefinitely improve autonomous driving planning performance in real-world scenarios (Section 5.1.2). This finding motivates us to explore data scaling strategies beyond simply augmenting data quantity. We examine how data distribution affects the efficiency of data scaling law (Section 5.2). Finally, we demonstrate that end-to-end autonomous driving systems, when scaled appropriately with data, exhibit combinatorial generalization capabilities.

5.1. Unveiling of Data Scaling Laws

5.1.1. Power-law Relationship

Inspired by the power-law data scaling laws observed in large-scale language models [16], we aim to investigate the existence of power-law scaling relationships in imitation learning-based end-to-end autonomous driving. We begin our analysis by examining the Average Displacement Error

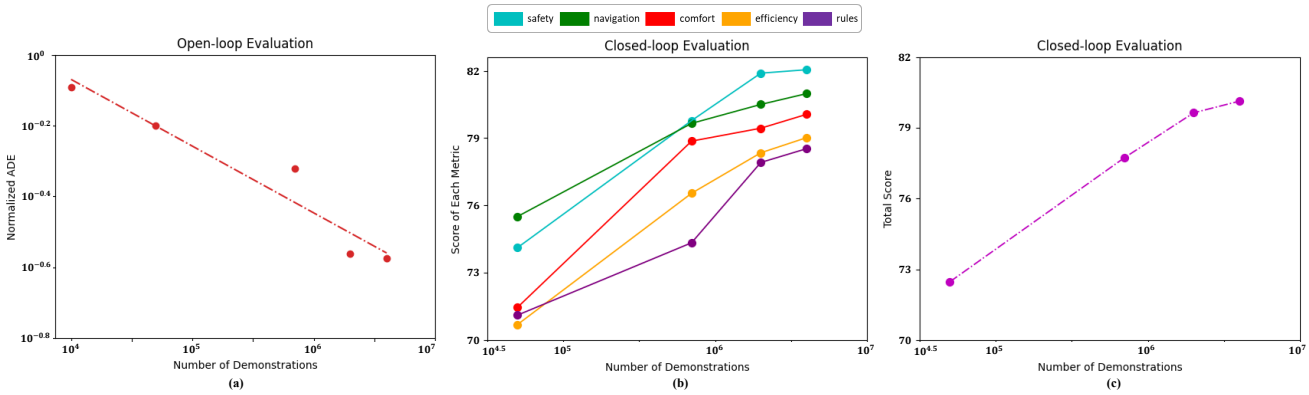

Figure 4. Results of open-loop [(a)] and closed-loop [(b) and (c)] evaluation. **(a):** Data scaling laws on ADE. Dashed lines represent power-law fits, with the equations and coefficient r provided in the eq. 2. All axes are shown on a logarithmic scale. **(b):** Scores of each metric in closed-loop evaluation. **(c):** Total score of closed-loop evaluation. In (b) and (c), the X-axis (number of demonstrations) is displayed on a logarithmic scale and the Y-axis is displayed on a linear scale.

(ADE) between ground truth trajectories and predicted trajectories in different training dataset scales. Specifically, we utilize open-loop evaluation results from five models with varying training dataset scales, as detailed in Section 3. We designate the amount of training data (quantified by the number of demonstrations used in training) as variable X and the normalized trajectory ADE as variable Y. After applying the logarithmic transformation to both X and X, the presence of a linear relationship would indicate a power-law scaling law in end-to-end autonomous driving data.

As shown in Figure. 4 (a), we conduct the linear model fitting of the log-transformed data. The fitting results yield:

$$Y = 0.6833 \cdot X^{-0.188}, r = -0.963 \tag{2}$$

The correlation coefficient r of the fit is -0.963, strongly suggesting the existence of a power-law data scaling law in trajectory fitting for imitation learning-based end-to-end autonomous driving.

5.1.2. Differences Between Open-loop And Closed-loop

Given that open-loop evaluation cannot fully reflect a vehicle's planning performance in real-world scenarios, we further assess the data scaling laws in a closed-loop condition. Specifically, we employ a visual closed-loop simulator to evaluate the vehicle's closed-loop performance of the models trained on variant scales of data. Following the description in Sec 4.3, we obtain the five closed-loop scores and the total score as illustrated in Figure. 4 (b) and (c).

Through the figure, we observe that nearly all of the closed-loop metrics have a similar relationship with the amount of training data. The score initially increases rapidly with the increase of the data scale, but after reaching an inflection point (about 2 million training demonstrations), the growth rate decelerates as data continues to expand. Surprisingly, this phenomenon differs significantly from those in open-loop assessment.

5.2. Data Quantity and Model Performance

Based on the foundation of data scaling law, we explore the relationship between data quantity and model performance when increasing the scale of training data. Specifically, we categorized our dataset of 2 million demonstrations into 23 scenario types, as defined in Section 4.2, and analyzed the proportion of each type. To simplify the process, we identify two scenario types that exhibit poor performance (DOWN_SUBROAD_LC and SUBROAD_TO_MAINROAD_HIGHWAY) from the 23 categories for in-depth study. By incrementally increasing the representation of these scenario types, we observe the consequent effects on planning performance.

As illustrated in Table. 3, we maintained a constant overall volume of training data while increasing the quantity of these two scenario types in each iteration. Subsequently, we evaluated the model's open-loop trajectory error after training. The results reveal that for these long-tailed scenarios, doubling the specific training data while keeping the total data volume constant led to improvements in planning performance ranging from 9.7% to 16.9%. Furthermore, quadrupling the specific training data yielded even more substantial gains, with performance improvements between 22.8% and 32.9%. Our findings suggest that even a relatively small increase in scenario-specific data (comprising only several hundred or thousand demonstrations) can lead to significant enhancements in planning performance for these challenging scenarios.

5.3. Combinatorial Generalization

In this section, we investigate the relationship between data scaling laws and the generalization capability, which is considered essential for deploying autonomous driving in real-

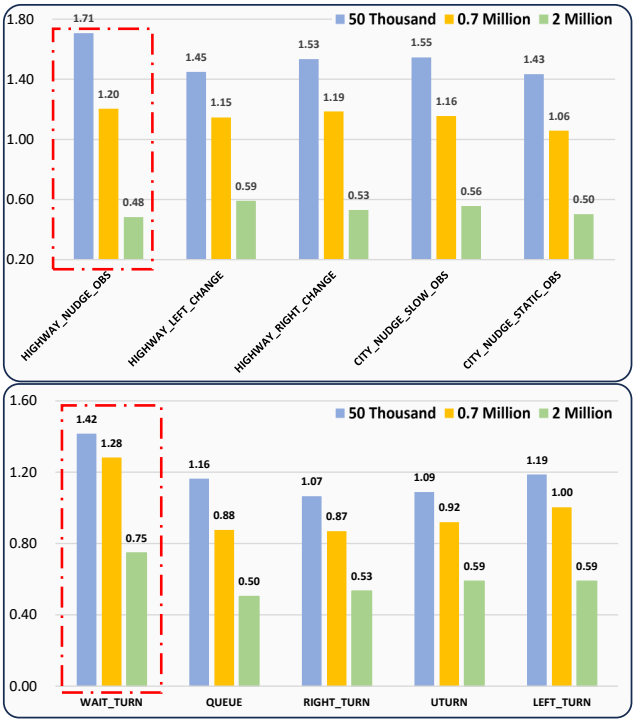

Figure 5. To investigate the relationship between data scaling and generalization ability, we select data from 2 scenario types, HIGHWAY_NUDGE_OBS and WAIT_TURN as test data. Models trained with 50 thousand, 0.7 million, and 2 million demonstrations are represented in blue, yellow, and green, respectively.

world scenarios.

5.3.1. Quantitative Analysis

To demonstrate the existence of generalization, we split two categories (HIGHWAY_NUDGE_OBS and WAIT_TURN) from the 23 scenario types defined in Sec 4.2 as test categories, while using the remaining 21 categories to form the training set. To be consistent with previous settings, we trained models using three scales of number of training data: **50 thousand**, **0.7 million**, and **2 million** demonstrations. Notably, we applied a strict filtering strategy when selecting test data for these two categories to ensure that each scenario does not overlap with other scenario types. The open-loop test results are shown in Figure. 5. For a convenient comparison, we select scenarios similar to these two scenario types.

We observed that (1) the model trained on 50 thousand demonstrations showed larger displacement error from expert trajectories in the two test scenarios compared to similar scenarios, indicating insufficient generalization capability with small-scale training data. (2) As the training data gradually increased to green demonstrations, the displacement error between the trajectories of the two test scenarios and other scenarios rapidly narrowed. The performance on HIGHWAY_NUDGE_OBS even surpasses the performance on other scenarios that participated in the training. (3) By learning high-speed driving and low-speed nudging obstacles separately from the training data, the model acquired the ability to generalize to high-speed HIGHWAY_NUDGE_OBS scenarios; through learning to turn and queuing at red lights, the model developed the capability to generalize to WAIT_TURN scenarios. Based on these observations, we argue that **the data scaling law endows models with the ability to combinatorial generalization.**

5.3.2. Qualitative Analysis

To demonstrate the generalization ability more intuitively, we also provide the qualitative visualization of the predicted semantic occupancy in Figure. 6, which shows the planning results of HIGHWAY_NUDGE_OBS and WAIT_TURN scenarios. The BEV maps in the green boxes represent the planning result of the model trained using 50 thousand demonstrations without HIGHWAY_NUDGE_OBS and WAIT_TURN scenarios (*50K + Unseen* in the legend), the BEV maps in the blue boxes represent the planning result of the model trained using 2 million demonstrations without HIGHWAY_NUDGE_OBS and WAIT_TURN scenarios (*2M + Unseen* in the legend), and the BEV maps in the orange boxes represent the planning result of the model trained using 2 million demonstrations with HIGHWAY_NUDGE_OBS and WAIT_TURN scenarios (*2M + Seen* in the legend).

Visual analysis reveals that appropriately increasing the scale of training data enables the model to achieve combinatorial generalization to novel scenarios. This enhanced generalization capability allows the model to perform competitively with counterparts specifically trained in these new scenarios. Our findings underscore the critical role of data scaling in improving model adaptability and robustness across diverse autonomous driving contexts.

6. Conclusion And Limitation

6.1. Conclusion

In this paper, we delve into the data scaling laws of the imitation learning-based end-to-end autonomous driving framework. Upon further investigation, we uncovered three intriguing findings:

- A power-law data scaling law in open-loop metric but different in closed-loop metric.
- Data distribution plays a key role in the data scaling law of end-to-end autonomous driving.
- The data scaling law endows the model with combinatorial generalization, which powers the model's zero-shot ability for new scenarios.

Table 3. Analysis of different scene distribution.

Scenario Type	Data Quantity			ADE ↓		
DOWN_SUBROAD_LC	4972	7218 (+45.1%)	21836 (+339.2%)	0.788	0.711 (−9.7%)	0.529 (−32.9%)
SUBROAD_TO_MAINROAD_HIGHWAY	643	1485 (+130.9%)	2569 (+299.5%)	0.786	0.653 (−16.9%)	0.593 (−22.8%)

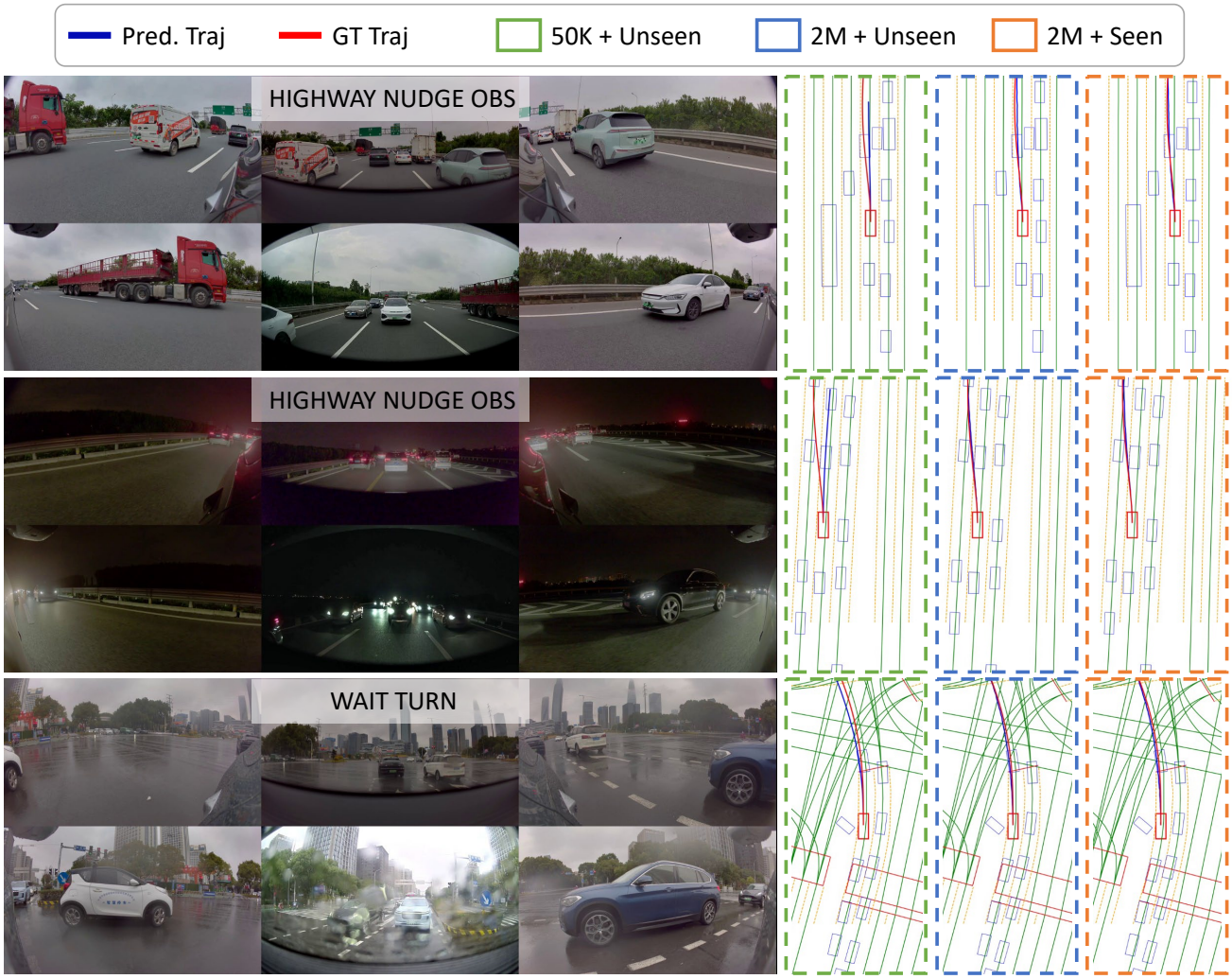

Figure 6. Qualitative results on scenarios of HIGHWAY_NUDGE_OBS and WAIT_TURN scenario types. We compare the planning trajectories generated by three different models, demonstrating the combinatorial generalization in new driving scenarios.

6.2. Limiataion And Future Work

While our work provides significant insights, we acknowledge several limitations. Due to computational constraints, our investigation primarily focused on imitation learning methods based on mean regression with Bird's Eye View (BEV) representation. To address these limitations and further enrich our understanding of data scaling laws in autonomous driving, we propose the following directions for future research:

- exploring scaling laws for a broader range of scene representations (BEV and sparse), model architecture (cascaded and parallel), and supervision paradigm (supervised and self-supervised);
- investigating the underlying factors contributing to the discrepancy in data scaling laws between closed-loop and open-loop evaluations;
- revealing how strategic adjustments to data distribution can enhance performance in closed-loop deployments.

These avenues could provide valuable insights into the scalability and generalizability of autonomous driving systems across diverse scenarios and model paradigms.

Preliminary Investigation into Data Scaling Laws for Imitation Learning-Based End-to-End Autonomous Driving

Appendix

Figure 1. Camera positions on the data collection vehicle.

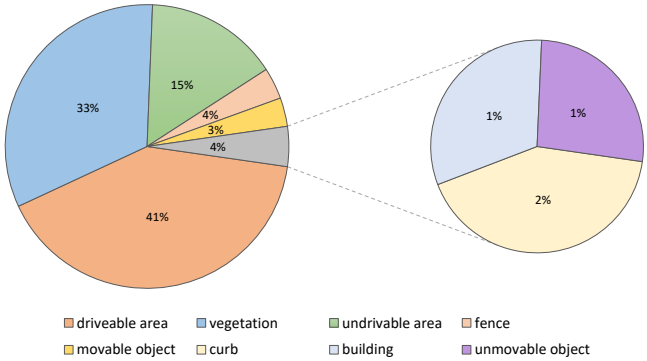

Figure 2. Distribution of each 3D occupancy category.

Table 1. Detailed hyper-parameters.

Config	Value
Image resolution	512×960
BEV resolution	232×80
Optimizer	AdamW
Learning rate	2e-4
Learning rate schedule	cosine
Batch size per GPU	8

1. Data Source And Annotation

1.1. Data Collection

We collect driving demonstrations from 14 cities, including Beijing, Shanghai, and Hangzhou, et al. The images are captured horizontally by seven cameras covering a 360° field of view FOV, as shown in Fig.1.

1.2. Annoatation

To train our model, we annotate the collected raw data with scene occupancy, HD maps, and motion information for both the ego vehicle and other agents. Similar to the nuPlan[5] dataset, we employed an automated pre-annotation process using models, followed by manual verification and refinement of the annotations by human annotators.

The HD map annotations include four categories: *lane dividers*, *lane centerlines*, *crosswalks*, and *stop lines*. The occupancy annotations comprise eight classes: *driveable area*, *vegetation*, *undrivable area*, *fence*, *movable object*, *curb*, *building*, *unmovable object*. The distribution of each 3D occupancy category is shown in Fig. 2.

2. Driving Scenarios Information

The specific characteristics of the 23 scenario types are demonstrated in Fig. 3 and Fig. 4.

3. Training Policy

Our experimental setup involved training models on five datasets of increasing scale, ranging from 10,000 to 4 million demonstrations (specifically: 10 thousand, 50 thousand, 0.7 million, 2 million, and 4 million demonstrations). We employed a uniform downsampling strategy to ensure consistency in data distribution across varying dataset sizes. Specifically, we divide the largest dataset containing 4 million demonstrations into 23 scenario types according to the definition described in Sec. 2. We then uniformly downsample data from each scenario type to construct the other four smaller datasets.

The hyperparameters used in the training process are presented in Tab. 1

Figure 3. Specific characteristics of the 23 scenario types (Part 1).

Figure 4. Specific characteristics of the 23 scenario types (Part 2).

4. Closed-loop Evaluation

4.1. Closed-loop Simulator Setup

To achieve high-fidelity simulation of real-world driving scenarios, we employ 3D Gaussian Splatting [17] (3D-GS) to reconstruct partial scenes for building a closed-loop simulator. Specifically, we utilize StreetGaussian [35] as our baseline method for scene reconstruction. To enhance ren-

dering quality from novel viewpoints, we collected data for each test scenario (with driving durations of approximately 10–30 seconds) using the same data collection vehicle traversing three distinct trajectories, each separated by 3 meters. Data is captured at a frequency of 10 Hz. For each trajectory, every frame is captured from 7 different viewpoints (the same as the process of collecting the training dataset; see Sec. 1.1 for details), covering comprehensive panoramic information.

The closed-loop simulation operates at a frequency of 10 Hz. During simulation, the ego vehicle follows the trajectory generated by our model, while other agents" motions are replayed from recorded trajectories.

4.2. Closed-loop Metrics

Closed-loop evaluation contains five metrics: safety, rule, navigation, efficiency, and comfort.

Safety. We measure the safety of the ego-planning based on the *no-at-fault collisions* and the *drivable area compliance*.

Rules. We measure the rules of the ego-planning to assess traffic rule compliance. Specifically, we implement penalties for five common traffic violations: crossing solid lines during lane changes, running red lights, failing to yield to pedestrians, driving against traffic, and speeding.

Navigation. We measure the rules of the ego-planning to assess navigation compliance. We implement penalties for violation of driving command.

Efficiency. We measure the efficiency of the ego-planning based on the *Ego Progress* metric of nuPlan [5]. We further extend the measurement to include lane changes and detours. In computing our metrics, we incorporated penalties for inefficient driving behaviors such as invalid lane changes and unnecessary detour initiations.

Comfort. We measure the comfort of the ego-planning similar to the strategy of nuPlan [5]. Specifically, we compare the minimum and maximum absolute values of lateral and longitudinal acceleration, the maximum absolute value of yaw rate, and the maximum value of turning radius with the comfortable thresholds.

4.3. Limitation

The simulator setup is only based on 3D-GS. When the ego vehicle's trajectory deviates significantly from the reconstructed data collection area, image rendering quality deteriorates noticeably. This limitation hindered our ability to evaluate a more diverse range of planning trajectories. In the future, advancements in reconstruction techniques or combining with generative models could potentially address this limitation.

Table 2. Occupancy Evaluation (Metric: mIoU).

	50 thousand	0.7 million	2 million
unmovable object	34.20	41.11	42.45
movable object	57.86	62.27	58.34
fence	51.04	55.38	60.81
undriveble area	61.03	64.33	69.78
driveble area	76.94	76.08	80.65
vegetation	44.46	51.20	48.51
building	39.25	46.65	48.18
curb	42.65	47.30	53.56
mIoU	50.93	55.54	57.79

Table 3. Online Mapping Evaluation (Metric: mAP).

	50 thousand	0.7 million	2 million
lane divider	61.02	65.11	74.62
lane centerline	70.05	72.07	80.16
crosswalk	50.21	55.13	63.60
stopline	76.91	79.17	86.67
mAP	64.55	67.87	76.26

5. Additional Experimental Results

5.1. Perception Results

As shown in Tab. 2 and Tab. 3, we evaluate the 3D occupancy and online map as perception results.

5.2. Real World Deployment

In a real-world deployment, we evaluate each model's miles per intervention (MPI), indicating the average distance traveled before human intervention is required. A higher MPI indicates better performance. Notably, in real-world applications, a safety check module is employed to select the most reasonable trajectory based on the perception results generated by the parallel decoder. Through training on 4 million demonstrations, our model achieved an average MPI of approximately 24.41 kilometers in road tests.

Bibliography

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 3

[2] Anonymous. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. under review. 1, 2

[3] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboa-

gent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024. 3

[4] 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 multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1, 2

[5] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 1, 2, 4

[6] Anh-Quan Cao and Raoul De Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3991–4001, 2022. 4

[7] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2022. 2, 3

[8] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023. 3

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 1, 3

[10] 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. 5

[11] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. 1, 3

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4

[13] 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. 2, 3, 4

[14] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *arXiv preprint arXiv:2406.03877*, 2024. 2

[15] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023. 2, 3, 4

[16] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1, 3, 5

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 5, 3

[18] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 3

[19] 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. 1, 2

[20] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14864–14873, 2024. 3

[21] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv preprint arXiv:2208.14437*, 2022. 4

[22] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024. 3

[23] Jingzhe Liu, Haitao Mao, Zhikai Chen, Tong Zhao, Neil Shah, and Jiliang Tang. Neural scaling laws on graphs. *arXiv preprint arXiv:2402.02054*, 2024. 3

[24] Junyi Ma, Xieyuanli Chen, Jiawei Huang, Jingyi Xu, Zhen Luo, Jintao Xu, Weihao Gu, Rui Ai, and Hesheng Wang. Cam4docc: Benchmark for camera-only 4d occupancy forecasting in autonomous driving applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21486–21495, 2024. 4

[25] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 3

[26] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 3

[27] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th Euro-*

pean Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, pages 194–210. Springer, 2020. 3, 4

[28] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023. 3

[29] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, pages 726–737. PMLR, 2023. 2, 3

[30] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024. 2, 4

[31] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024. 1, 3

[32] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023. 3

[33] 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. 3, 4

[34] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022. 3

[35] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. *arXiv preprint arXiv:2401.01339*, 2024. 3

[36] Jiang-Tian Zhai, Ze Feng, Jinhao 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 nuscenes. *arXiv preprint arXiv:2305.10430*, 2023. 3

[37] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022. 3

[38] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024. 3