LEAD: Minimizing Learner-Expert Asymmetry in End-to-End Driving

Long Nguyen[1] Micha Fauth[1] Bernhard Jaeger[1] Daniel Dauner[1]
Maximilian Igl[2] Andreas Geiger[1] Kashyap Chitta[1,2]
[1]University of Tübingen, Tübingen AI Center [2]NVIDIA Research

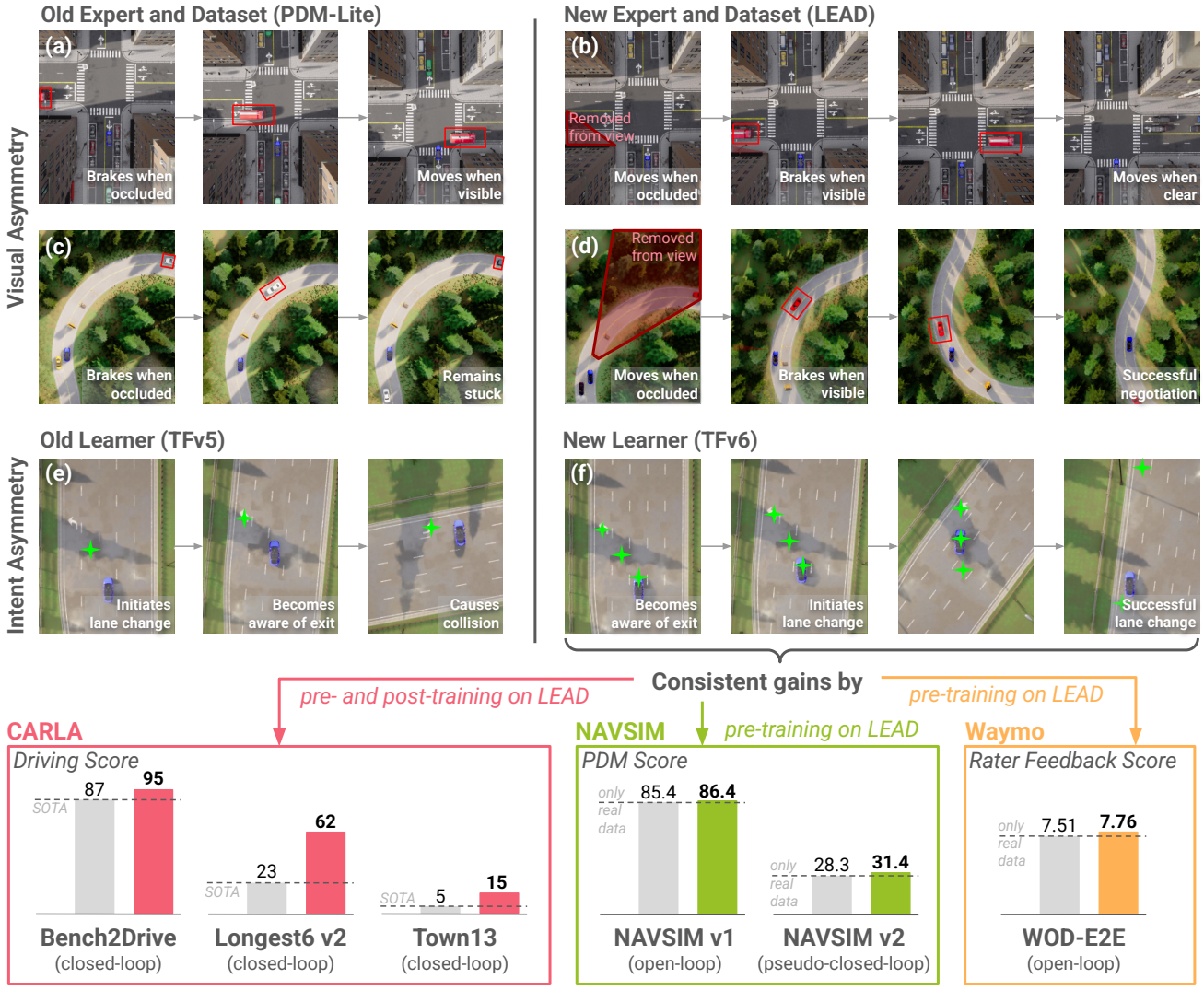

Figure 1. **(Top) Learner-Expert Asymmetry.** (a,c) Unrealistic expert demonstration: the expert (blue) stops before the fire truck or oncoming vehicle (red boxes) are in the field of view. (b,d) Our improved expert (LEAD) explicitly reasons about occlusions, thereby behaving more realistically: It only stops once the vehicles are visible, leading to explainable and more challenging behavior (e.g. it now needs to negotiate with the oncoming vehicle in (d)). (e) Insufficient route information (single green star) in the old learner policy can lead to undesired behavior. (f) By conditioning on multiple target points, our new learner executes a lane change successfully. **(Bottom) Performance Across Benchmarks.** Using our high-quality synthetic data, we train imitation learning policies that achieve state-of-the-art performance on closed-loop simulations (CARLA) and improve results on benchmarks using real-world data (NAVSIM and Waymo).

Abstract

Simulation-generated datasets for autonomous driving rely on omniscient data collection 'expert' policies, which use unobservable scene information (e.g., from occluded regions) to make driving decisions. When such data is used for end-to-end policy training, it results in an information asymmetry between the expert and the 'learner' policy, which has limited sensor coverage and navigational intent information compared to the expert. We show that this asymmetry leads to a significant drop in the performance of the learner. To combat this, we present LEAD, a new high-quality synthetic dataset collected in the CARLA simulator with three key improvements. (1) The expert minimizes its use of unobservable information by removing entities from its input state that would be occluded in the learner's field of view. By providing the learner with (2) detailed driver intent information and (3) rich sensor modalities (cameras, LiDARs, radars, and odometry), the dataset narrows down the information gap between the learner and expert. We then propose TransFuser v6 (TFv6), a simple end-to-end learner policy trained on LEAD. As a result of our improvements, TFv6 substantially advances the state of the art on all publicly available CARLA closed-loop driving benchmarks, reaching driving scores of 95 on Bench2Drive, 62 on Longest6 v2, and 15 on the Town13 validation routes. Finally, we aggregate the LEAD dataset with several public real-world datasets under a unified repository to enable cross-dataset evaluation. We show that pre-training TFv6 on synthetic data from LEAD leads to consistent performance gains when followed by fine-tuning with real data from the NAVSIM v1/v2 and WOD-E2E benchmarks.

1. Introduction

Learning by Cheating (LBC) has proven to be an effective paradigm for tackling vision-based robotics and autonomous driving tasks [6]. LBC works in two distinct phases. First, we train or program an 'expert' to drive while 'cheating' by giving it access to ground-truth information, e.g., the exact map layout and the precise positions of all other traffic participants. This agent performs the planning sub-task without the burden of perception. Next, we train a 'learner' (the final model) to imitate the privileged expert's actions. However, this final agent uses only sensor inputs (e.g., camera images) and must therefore learn perception sufficiently well to replicate the expert's decisions. A rich body of recent literature and various datasets collected by privileged experts in the CARLA simulator [18], such as PDM-Lite [47], Bench2Drive [27], DriveLM-CARLA [48], and SimLingo [42], support this methodology.

However, these datasets take the expert's privileged nature to the extreme: they provide experts with inputs that extend far beyond solving perception. The expert hence possesses omniscient dynamics understanding and a perfect Field of View (FOV), spanning several hundred square meters around the vehicle, whereas the learner must rely on low-resolution, partial-FOV sensors. For example, PDM-Lite, the current state-of-the-art expert, often utilizes information from occluded areas in its decision-making, e.g., braking for a passing vehicle before it is visible (Fig. 1 (a-d)). Furthermore, the expert accesses the long-term routing intent spanning several hundred meters ahead of its current position, while the learner sees only a single target point at a random distance of 7.5–200 m away (Fig. 1 (e)). We name these mismatches in terms of visibility and intent the *learner-expert asymmetry*.

Surprisingly, most prior work has overlooked this problem, focusing on algorithmic improvements rather than dataset biases. In this paper, through systematic analysis, we observe that both visual and intent asymmetry lead to major degradation in learner performance. To tackle the visual asymmetry problem, we constrain expert visibility during synthetic data collection. The expert remains privileged, using bounding boxes to represent traffic agents, but we filter agents from its input when they are not in the learner's FOV. To tackle intent asymmetry, we improve the learner's input space rather than constraining the expert. We achieve this by collecting a three-point representation of the local route (previous, current, and next target point) to condition the model, instead of using only a single target point. To further improve the learner's input space, we collect data with a comprehensive sensor suite. Specifically, we include radar sensors which are rarely used in end-to-end driving datasets yet provide crucial information regarding the velocities of other agents. Combining these improvements, we collect a new dataset, LEAD, which is designed to minimize Learner-Expert Asymmetry in Driving.

We then propose TransFuser v6 (TFv6), a low-latency, simple, and general learner policy architecture that supports flexible formats for both intent and sensor inputs. TFv6 achieves exceptionally high performance on every existing CARLA benchmark when trained on LEAD, with each of our proposed dataset improvements responsible for a significant proportion of the performance gain. We evaluate TFv6 on Bench2Drive, a popular benchmark featuring 50 recent methods from the past two years [27]. Our approach achieves a driving score 8 points higher than the current state of the art (SOTA), a significant leap in a benchmark where top methods are usually separated by only 1-2 points. On Longest 6 v2, we more than double the SOTA performance, raising it from 23 to 62 [26]. Similarly, on Town 13, the most challenging closed-loop benchmark on CARLA, we triple the current SOTA performance from 5 to 15 [59].

Our TFv6 architecture's simplicity allows us to co-train on multiple datasets despite minor differences in sensor and

intent input formats. We aggregate LEAD with popular real-world datasets into a unified repository that enables cross-dataset training and evaluation. On NAVSIM [17], pre-training TFv6 on data from LEAD leads to improved performance after fine-tuning on in-domain data, particularly on the challenging `navhard` benchmark [3] which tests the ability to recover from perturbations. We observe similar results on the Waymo Open Dataset End-to-End driving benchmark consisting of rare long-tail scenarios [54], thus providing initial evidence of the benefits of high-quality synthetic data in end-to-end AV development. Our code and data will be made publicly available.

2. Related Work

End-to-End Driving via Imitation Learning: Imitation Learning (IL) can be used to train a neural network policy end-to-end, taking in sensor data and predicting an action representation of expert behavior by minimizing the difference between predicted and observed actions on an offline dataset. Invented in 1988 [40] and revived in 2016 [2], IL has become the predominant paradigm for end-to-end driving today [8, 28, 36, 37, 42, 49, 59], following a period of rapid progress on public driving benchmarks [5–7, 9, 10, 13, 14, 22, 25, 41, 45, 46, 52]. End-to-end driving methods can be trained with human data and evaluated with open-loop metrics on static datasets [3, 17, 22, 54]. However, open-loop metrics have been observed to be unreliable [12, 35, 51, 56] due to driving being a closed-loop task. Benchmarking closed-loop driving is possible in simulators [18, 32, 38, 44, 53, 55, 58]. In this work, we use the most popular and feature-rich autonomous driving simulator, CARLA [18], which is built upon Unreal Engine.

Human-annotated training data is typically too costly to acquire in simulations. Instead, learning by cheating is used ubiquitously across the literature [6]. In this approach, a privileged rule-based [1, 13, 16, 23, 48] or reinforcement learning-based [15, 26, 33, 57] planner, called an expert driver, with access to the ground-truth simulator state, is first created and collects and annotates the data for subsequent imitation learning. It is well recognized that the quality of driving behavior the expert driver provides is important and imposes an upper bound on the performance of the IL agent. However, besides the performance of the expert, we show that the alignment of the expert and IL agent input space is largely overlooked. Aspects that influence the decision-making of an expert should be present in the learner's observation space. In our study, we build upon the rule-based PDM-Lite expert [48], which, unlike Think2Drive [33], is open-source and enables us to analyze influential factors for expert-learner alignment.

Learner-Expert Visual Asymmetry: The quality of demonstrations is a critical bottleneck for the performance of IL. Chitta et al. [10] improved a prior rule-based driving expert to leverage more privileged information, including precise agent positions, velocities, and future intentions, which improved both expert and student driving performance. Zimmerlin et al. [59] demonstrate that simply improving expert performance does not guarantee high-quality supervision to a student. For example, experts may react to information unavailable to the student policy, such as slowing down for pedestrians outside the camera or LiDAR field of view. During training, the policy observes the deceleration but not its cause, breaking the causal link between perception and action. In this work, we thoroughly investigate such visual asymmetries and demonstrate that addressing them through aligned data collection substantially improves learned policy performance.

Learner-Expert Intent Asymmetry: Codevilla et al. [13] show that explicit navigation conditioning is essential for E2E driving policies to resolve ambiguity at intersections, where identical visual inputs must produce different actions based on intended routes. However, Jaeger et al. [25] further identified that models can over-rely on navigation signals when perceptual representations are weak, using target points as shortcuts that bypass perception rather than as navigational guides. In our work, we systematically study the conditioning mechanism for route information, demonstrating that careful design choices substantially improve both safety and robustness.

3. Minimizing Learner-Expert Asymmetry

This section presents controlled experiments that show how minimizing the learner-expert asymmetry can improve the performance of a strong baseline learner policy.

3.1. Preliminaries

We consider the task of navigating through urban scenarios along a predefined route. Each route is represented by a sequence of sparse GNSS coordinates, called target points.

Benchmark: We use the CARLA simulator version 0.9.15 and the longest6 v2 benchmark routes [26]. On longest6 v2, the agent is tested on 36, 1-2 km long routes in town 1-6, which include 6 predefined safety-critical scenario types.
Metrics: We use the official CARLA closed-loop metrics in our experiments. The Route Completion (RC) is the percentage of the completed route, whereas the Infraction Score (IS) is a factor starting at 1.0 that decays with every misbehavior. The Driving Score (DS) forms the primary metric by multiplying RS and IS.

Baseline: Our approach builds on top of *TransFuser++* [25, 59]. To avoid confusion due to the large number of TransFuser variations, we adopt the versioning nomenclature [24] and call this model **TFv5** from now on. TFv5 is an end-to-end driving model trained with imitation learning

TFv5 trained with...	DS ↑	RC ↑	Stat↓	Ped↓	Veh↓	OL↓	Red↓	Dev↓	Stop↓	Block↓
PDM-Lite dataset [48]	22.51 ± 4.42	70.68 ± 8.24	0.46	0.05	1.37	0.46	0.58	0.16	0.18	0.27
LEAD dataset (Ours)	**34.05** ± 1.50	**62.68** ± 11.30	**0.33**	**0.02**	**0.65**	**0.24**	**0.05**	**0.10**	**0.05**	**0.24**

Table 1. **Visibility Alignment**. We compare TFv5 (ResNet-34) performance when trained on PDM-Lite versus our proposed LEAD dataset. Dataset improvements alone yield a $+11.5$ point gain in Driving Score (DS), primarily through reducing infraction penalties. **Stat**: Collisions with Layout; **Ped**: Collisions with Pedestrian; **Veh**: Collision Vehicle; **OL**: Outside Lane; **Red**: Red Light; **Dev**: Route Deviation; **SI**: Stop Infraction; **Block**: Vehicle Blocked. All auxiliary metrics are normalized by kilometers driven.

(IL) that applies separate CNN encoders on a large front-view RGB image and a LiDAR raster, with intermediate transformer layers to fuse both modalities [41]. Multiple prediction heads operate on the encoded input in parallel, including auxiliary perception decoders [10], a GRU-based path decoder [25], and a discrete target speed prediction [25]. The vehicle is controlled by steering towards the predicted path using a PID controller and by controlling longitudinal velocity using a linear regression model that aims to track the predicted target speed. TFv5 is a strong baseline with close to state-of-the-art performance on Bench2Drive [27], longest6 v2, and the CARLA leaderboard 2.0 validation and test routes. We base our method on it because it requires less inference compute than competing approaches [42, 49], so it offers the best performance-efficiency trade-off. In the following sections, we make several improvements to TFv5, leading to our new method TFv6.

3.2. Visibility Alignment

TFv5 is trained with data that is collected and automatically labeled by the privileged expert planner *PDM-Lite* [48]. PDM-Lite is a rule-based planning method that incorporates classic planning ideas [50] and model-predictive control strategies. It was optimized for maximum performance on the CARLA leaderboard 2.0. However, as we highlight in this section, good planning performance does not imply that PDM-Lite is best suited as an expert for imitation learning policies. In particular, any information mismatch between the inputs to the expert driver and the learned model, e.g. caused by occlusions, can lead to causal confusion in the learner. Specifically, PDM-Lite uses the bounding boxes of all surrounding agents as input, whereas TFv5 does not see vehicles or pedestrians that are occluded by the static environment. Consequently, when PDM-Lite reacts to occluded vehicles, TFv5 is forced to associate the driving decision with other patterns in its input, leading to causal confusion [8].

To prevent this, we incorporate an occlusion check into PDM-Lite, where we exclude vehicles and pedestrians from PDM-Lite's input that are not visible in the camera of the downstream agent. This is achieved using CARLA's instance segmentation camera together with depth unprojection to reconstruct a dense point cloud aligned with the camera view. For each actor, we then estimate its visibility by counting the number of pixels within its bounding box that are actually seen by the camera.

A second mismatch is the range over which objects PDM-Lite reasons. It can see all objects that are up to 96 meters away. TFv5 uses a single front-view camera combined with a surround-view LiDAR. To account for objects only visible in the LiDAR, we include objects visible in the LiDAR and exclude all objects that fall outside the LiDAR range. The LiDAR of TFv5 has a smaller effective perception range than PDM-Lite, which is rasterized into a grid of size 64×96 meters.

Using this expert, we collect *LEAD*, a new dataset gathered on the same routes as PDM-Lite and comparable in size. Table 1 compares TFv5 trained on *LEAD* versus on PDM-Lite. TFv5 achieves an improvement of 12 DS on the longest6 v2 benchmark with this new dataset, demonstrating that the visual input mismatch between expert and learner was a significant issue. Table 3 shows that LEAD and PDM-Lite achieve almost identical driving scores. This shows that the improvement is driven by minimizing the learner-expert asymmetry and not by having a stronger-performing expert. Note that this improvement comes at no computational cost in terms of training or inference time.

3.3. Intent Alignment

To drive well, IL methods need to be conditioned on the high-level goal of where to drive to disambiguate the correct action at intersections [13]. Modern end-to-end driving architectures that are tested on long routes in closed-loop benchmarks are conditioned using a target point GNSS location. Besides conditioning, this target point is implicitly used by models to recover from compounding errors [25]. They blindly drive towards the next target point when out of distribution, which returns them to in-distribution states.

The CARLA leaderboard 2.0 benchmarks introduces static obstacles, which enables measuring a new downside of the target point bias. Fig. 2 (a) shows an example where TFv5 produces unsafe trajectories that pass too close to static vehicles. This slows ego progress and increases the risk of a collision with oncoming traffic. To combat this problem, we make several changes to the target point conditioning. First, we remove the GRU, following [20, 42], and add the target point as an encoded token to the transformer decoder, moving it further away from the output in the net-

LEAD used to train...	DS ↑	RC ↑	Stat↓	Ped↓	Veh↓	OL↓	Red↓	Dev↓	Stop↓	Block↓
TFv5 [59]	34.05 ± 1.50	62.68 ± 11.39	0.33	0.02	0.65	0.24	**0.05**	**0.10**	0.05	0.24
TFv6 (Ours)	**42.13** ± 0.75	**62.87** ± 2.07	**0.01**	**0.00**	**0.42**	**0.12**	**0.05**	0.33	**0.03**	**0.01**

Table 2. **Intent Alignment**. We compare the TFv5 [59] against TFv6 (ours) when trained on the LEAD dataset. Our intent alignment and improved route conditioning substantially improves the Driving Score (+8) and infraction penalties, with similar route completion rates.

Figure 2. **Target Point Bias.** a) TFv5 outputs trajectories (red dots) and spatial paths (blue line) which are heavily influenced by the target point (green star). In situations such as overtaking static vehicles, this results in unrealistic and unsafe trajectories that pass dangerously close to the static vehicles. b) With the modifications incorporated in TFv6, we observe that the model is less prone to this bias.

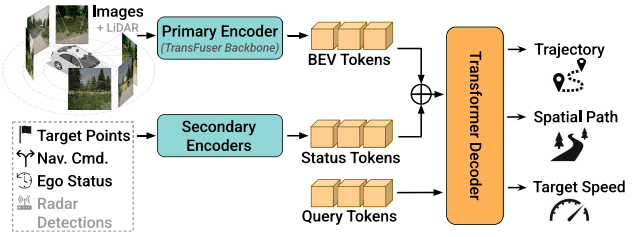

Figure 3. **TransFuser v6 Architecture.** We use the TransFuser [41] backbone to encode multi-view images into Bird's-Eye-View (BEV) tokens. Auxiliary inputs, such as target points, navigation commands, and past ego states, are processed by secondary encoders. Then, a transformer-based decoder [25] jointly predicts the future trajectory, spatial path, and target speed. If available, LiDAR and radar can be fused as additional sensor inputs.

work architecture. Second, instead of using just one target point as conditioning, we additionally condition on the last target point that was passed and the next target point. This is useful for higher driving speeds in CARLA leaderboard 2.0, where the model might need to predict waypoints that are farther away, past the next target point, which is ambiguous otherwise. We add trajectory prediction into the architecture as an auxiliary loss in CARLA, and as a primary output for benchmarks that do not contain target speed labels. Our TFv6 architecture is visualized in Fig. 3. Note that in this section we use the sensor setup of TFv5 for a fair comparison, and the extension of sensors is incorporated in Section 4.

Table 2 shows that these architectural changes together improve the models' driving score by +8 on longest6 v2. Qualitatively, we observe that the target point bias of the model is weaker after these changes. Fig. 2 (b), for example, shows that the new model now ignores the target point and successfully overtakes the static vehicle. This can be quantitatively seen in the lower collision metrics (Stat, Ped, and Veh). However, we also observe that without the target point bias, the model has weaker recovery ability as indicated by the increase in route deviation. This is not apparent in the route completion metric because our changes also reduce vehicle blocked infractions.

4. Experiments

In this section, we provide system-level comparisons of TFv6 on 5 different benchmarks against state-of-the-art methods and other baselines. Our main experiments involve closed-loop CARLA benchmarks, where TFv6 outperforms all prior work by a large margin. Additionally, we analyze different sensor setups, showing that a combination of cameras, LiDAR, and radars achieves the best performance. Lastly, we show that training on synthetic data improves

Method	Backbone	Input Modalities			Bench2Drive		Longest6 v2	
		Cameras	LiDAR	Radar	Driving Score ↑	Success Rate ↑	Driving Score ↑	Route Completion ↑
HiP-AD [49]	ResNet-50	6×	✗	✗	86.8	69.1	7	56
SimLingo [42]	InternViT-300M	1×	✗	✗	85.1	67.2	22	70
TFv5 [59]	RegNetY-032	1×	✓	✗	84.2	67.3	23 ± 4	70 ± 8
TFv6 (Ours)	ResNet-34	6×	✗	✗	91.6 ± 0.7	79.5 ± 2.0	43 ± 1	85 ± 3
		6×	✓	✗	94.7 ± 0.6	85.6 ± 0.0	52 ± 7	88 ± 5
		6×	✗	✓	94.2 ± 0.7	85.3 ± 0.9	52 ± 1	88 ± 2
		6×	✓	✓	95.0 ± 0.7	84.3 ± 2.1	54 ± 5	89 ± 3
		3×	✓	✓	94.7 ± 0.7	82.1 ± 3.6	57 ± 3	**99** ± 0
	RegNetY-032	3×	✓	✓	**95.2** ± 0.3	**86.8** ± 0.7	**62** ± 1	91 ± 1
PDM-Lite [48]	-		-		*97.0*	*92.3*	*73*	*100*
LEAD (Ours)	-		-		*96.8*	*96.6*	*73*	*93*

Table 3. **CARLA Bench2Drive and Longest6 v2.** TFv6 outperforms all baselines. Our best model uses three cameras, LiDAR and Radar.

Method	RC ↑	IS ↑	DS ↑	I ↑	NDS ↑
'Town13 Train' – Trained on all towns					
UniAD [22]	1.42	**0.49**	0.23	**0.30**	0.00
TFv5 [59]	68.53	0.04	0.96	0.07	4.94
TFv6 (Ours)	**71.82**	0.12	**5.01**	0.20	**14.65**
'Town13 Val' – Town13 withheld during training					
TFv5 [59]	**50.20**	0.10	1.08	0.04	2.12
TFv6 (Ours)	39.70	**0.28**	**2.65**	**0.22**	**4.04**
PDM-Lite [48]	*83.40*	*0.41*	*36.30*	*0.63*	*58.50*

Table 4. **Benchmarking on CARLA Town13.** Mean over 3 evaluations of each agent. *Results included for completeness, though this setting is not the recommended default for this benchmark.

performance on real-world open-loop benchmarks.

4.1. Closed-Loop Simulation Benchmarks

Benchmarks: We test TFv6 on three established benchmarks on the CARLA [18] simulator, Bench2Drive [27], longest6 v2 [26] and Town 13 validation [59]. **Bench2Drive** features 220 short, 50m-200m long routes with over 44 different scenario types across 12 towns. It tests how well a method can handle the diversity of different driving situations and is computationally cheaper to evaluate than the other benchmarks used in this work. Bench2Drive is very popular, with over 40 recent methods being tested on the benchmark. We show a comparison with every existing method in the supplementary material. However, the short length of the test routes means that Bench2Drive is not ideal for measuring long-term consistency or the ability to recover from compounding errors. To complement this, we additionally test our model on the **longest6 v2** benchmark, which features 36 1.0-2.5 km long routes with 6 types of scenarios in 6 towns. Longest6 v2 is therefore better able to measure recovery from compounding errors. Lastly, we evaluate on the **Town13 validation** benchmark. It consists of 20 routes that are on average 12.39 km long, featuring roughly 100 scenarios per route of 38 different types. Unlike the prior two benchmarks, all routes are in the huge Town 13, and methods are not allowed to train on data collected in that town, therefore testing the ability of methods to generalize to novel environments. Town 13 validation can be considered the most challenging autonomous driving benchmark today. The level of consistency needed to solve 100 safety-critical scenarios consecutively, in addition to generalizing to an unseen environment, is far beyond the capabilities of today's state-of-the-art methods.

Metrics: We use the standard metrics of each respective benchmark. **Success Rate** (SR) is the percentage of routes that have been completed without infraction. **Route Completion** (RC) measures the percentage of the route that was completed. **Driving Score** (DS) multiplies RC with the **infraction score** (IS), a penalty $\in [0, 1]$ that reduces multiplicatively every time the agent incurs an infraction. On town 13 validation, the DS has been observed to be unreliable due to scores of state-of-the-art methods being too low [59]. We additionally report the **Normalized Driving Score** (NDS) [59], which solves this problem by multiplying route completion by the infraction coefficient **I**, a variant of IS that is normalized by distance driven.

Training: We train TFv6 using a dataset collected with the LEAD expert. The dataset is larger than the one we used in Section 3, containing 73 hours of driving instead of 40 hours. We train TFv6 with 4xL40S GPUs for roughly 1 week in mixed-precision [39]. We use two-stage training [25], where the first stage only trains perception losses. Both stages are trained for 30 epochs. Further training details can be found in the supplementary material.

Radar: We explore the role of radar as an additional mid-range sensing modality within our TFv6 architecture. Although radar is widely used in production AV stacks, it remains almost entirely absent from academic end-to-end driving datasets and benchmarks. Prior CARLA datasets

mostly omit radar entirely. In contrast, LEAD provides four radar units, each emitting roughly 75 detections per frame containing 3D location and relative radial velocity, enabling us to systematically study their contribution.

In TFv6, each radar detection is encoded by sampling the BEV features at the detection location and concatenating them with its measured radial velocity as well as location. A small MLP maps this vector to a radar embedding, which, along with ego-velocity, serves as context to a fixed set of learned queries in a 4-layer transformer module. The detector is trained using a DETR-style matching loss [4], predicting object presence, bounding boxes, and velocity vectors, and is optimized jointly with the existing perception heads during perception pretraining.

During planning, the radar queries serve as an additional context input to the transformer decoder, providing explicit cues about moving agents and improving the model's ability to anticipate dynamic interactions that are partially occluded or hard to infer from cameras and LiDAR alone.

Baselines: **HiP-AD** [49] is a camera-only end-to-end driving approach and the current published state-of-the-art on Bench2Drive. **TFv5** [59] is an end-to-end driving method that fuses LiDAR and camera data with Transformers and the latest iteration of the classic TransFuser architecture [41]. **SimLingo** [42] is a recent vision language action model that ranks second on Bench2Drive. **UniAD** [22] is a popular end-to-end driving approach that sequentially utilizes auxiliary losses. **PDM-Lite** [48] is a privileged rule-based planning method that uses ground truth perception inputs and is used for automatic data collection.

Results: Table 3 shows that TFv6 outperforms TFv5 by 11 points in DS and by 20 points in SR on Bench2Drive. TFv6 sets a new state-of-the-art on Bench2Drive, outperforming the best published method, HiP-AD [49], by 8 DS and 18 SR. This is a significant leap in performance. In particular, because Bench2Drive is a mature benchmark with 48 baselines. We report the results of the other baselines in the supplementary. To set this improvement in perspective, HiP-AD outperforms SimLingo [42], the second-best published method, by 2 DS and 2 SR. TFv6 comes close in performance to its expert LEAD with only a 2 DS gap, although the SR gap is still 10 points.

On longest6 v2, the improvement over TFv5 is even more pronounced, with gains of +39 DS and +21 RC. Notably, the state-of-the-art Bench2Drive method, HiP-AD, performs poorly on this benchmark, achieving only 7 DS. Qualitative analysis shows frequent failures in which HiP-AD struggles to remain on the road for extended periods and becomes stuck after driving onto sidewalks.

These results underscore the importance of evaluating methods on long driving routes. This is particularly concerning given that CARLA is currently the only widely used simulator that supports long-form evaluation. Most recent benchmarks and simulation frameworks [3, 17, 21, 29, 30, 38, 54, 58] are limited to short-form driving due to log-replay or reconstruction-based designs that inherently prevent long-horizon testing. Generative approaches [11, 43] may offer a path forward in addressing this limitation.

Table 3 also shows several ablations on different sensor setups with TFv6. We find that using 3 cameras with a combination of LiDAR and Radar sensors yields the best results, +19 DS on longest6 v2 compared to using cameras only. Additionally, we show that the RegNetY-032 backbone provides a performance improvement of +5 DS on longest6 v2 compared to the smaller ResNet-34 backbone, consistent with the findings of [10].

Table 4 shows the results on the CARLA town 13 benchmark. TFv6 outperforms TFv5 by 1.9 NDS as well as 1.6 DS. Note that these results do not use early stopping [59] as indicated by the high route completions. TFv6 drives much more safely than TFv5 as indicated by its higher IS and I metrics, but also drives more conservatively as indicated by the lower RC. On Town 13 Validation, the gap to the privileged expert driver remains substantial. To highlight the impact of generalization, we also evaluate TFv6 trained with data from town 13 (Town 13 Train), but shade the results in gray to make it clear that these results are only for analysis purposes. We observe a large generalization gap. TFv6 achieves 14.65 NDS on Town 13 Train, which drops to 4.04 NDS when evaluated on Town 13 Val. This highlights the need for validation benchmarks where methods do not train on any data collected in the validation town. Many contemporary benchmarks do have training and testing areas cleanly separated by geographic region. Overall, TFv6 presents a substantial step forward for the state-of-the-art on Town 13 Val, but the gap to the privileged expert driver remains huge. The town 13 Validation benchmark was released already 3 years ago in 2022, but has not been used in any works up until Zimmerlin et al. in late 2024 [59]. The reason was that the difficulty of this benchmark was overwhelming for state-of-the-art methods. The popular approach UniAD [22], for example, achieves a trivial score of 0 NDS even disregarding generalization. Given that scores on many other benchmarks are saturating, it may be time for the community to revisit this challenge.

4.2. Real-World Data Benchmarks

We further supplement our evaluation with multiple real-world benchmarks for end-to-end driving: **(1) NAVSIM v1** [17] requires a sensor agent to plan a 4-second trajectory given multi-view camera observation, historical vehicle states, and discrete driving commands over a 2-second history. The benchmark is based on the nuPlan dataset [29] and specifically filters out ordinary driving situations with trivial solutions. An open-loop rollout of the trajectory is scored with the Predictive

Method	**NAVSIM v1** navtest PDMS ↑	**NAVSIM v2** navhard EPDMS ↑	**WOD-E2E** Validation RFS ↑
Ego Status MLP [17]	65.6	12.7	7.31
LTF [10]	83.8	23.1	-
LTFv6	85.4	28.2	7.51
+ LEAD Pre-Training	86.4	31.4	7.76
RAP [19]	93.7	39.6	-

Table 5. **LTFv6 on Real-World Data.** We show results on the navtest splits in NAVSIM v1, the navhard split of NAVSIM v2, and the validation split of WOD-E2E. Our LTFv6 model and LEAD training strategy exhibit consistent performance gains across benchmarks.

Driver Model Score (PDMS), which combines several sub-metrics, including no-collision, progress, time-to-collision. **(2) NAVSIM v2** [3] further extends v1 and introduces a two-stage evaluation pipeline and the Extended PDMS (EPDMS), that includes more sub-metrics, i.e., traffic-light compliance, lane keeping, and extended comfort. The second stage approximates a closed-loop rollout with pre-generated observations from 3D Gaussian Splatting [31, 34]. EPDMS weights the second stage outcomes based on the first stage endpoint proximity. **(3) WOD-E2E** [54] is an open-loop benchmark that involves predicting a 5-second trajectory, given similar multi-view camera images, vehicle states, and a driving command. The benchmark builds on 4,021 curated 20-second segments that specifically sample long-tail events occurring with less than 0.003% frequency in daily driving. The Rater Feedback Score (RFS) evaluates trajectories against three expert-annotated quality annotations ranging between 0 and 10. The agent either receives the expert score if the predicted trajectory falls within threshold-defined trust regions, or else the score is exponentially decayed.

Training: For the NAVSIM benchmarks, we use the full *navtrain* split and a subset of 100k frames from CARLA. For the CARLA data, we adapt the camera parameters to match NAVSIM and specifically sample scenarios that involve traffic rules and agent interactions. We only use the perception labels from the synthetic CARLA data to avoid the driving-style mismatch between human data collectors and LEAD. We train on mixed data in the first 30 epochs, which smoothly excludes the synthetic data, followed by 90 epochs excluively training on navtrain. Similarly, for WOD-E2E, we pre-train for 30 epochs exclusively on CARLA data, followed by 30 epochs of fine-tuning on the WOD-E2E training split. We provide further information in our supplementary material. Since LiDAR and radar data are not available in all the benchmarks, we drop these modalities and replace the LiDAR with a positional encoding, as done in the Latent TransFuser (LTF) variant [10]. We call the resulting method **LTFv6** to indicate that this version matches the TFv6 architecture.

Results: As shown in Table 5, we achieve notable improvements from our LTFv6 architecture compared to Latent TransFuser, with $+1.6$ and $+5.1$ in the NAVSIM v1 and v2 scores, respectively. Generally, LTFv6 improves in the ego progress, drivable area compliance, and traffic light compliance submetrics with minor trade-offs in comfort. Our proposed joint pre-training further increases the LTFv6 score consistently across all benchmarks, demonstrating the value of synthetic data despite the presence of distribution shifts. The training with LEAD improved our score by $+0.25$ in RFS. While the concurrent work RAP exhibits stronger performance overall [19], the method requires significantly more compute (e.g., using a 0.9B parameter backbone) and NAVSIM-specific optimization objectives. In contrast, our lightweight model demonstrates that improvements from simulation and co-training with synthetic data translate to real-world data.

5. Conclusion

In this paper, we identify and address a critical yet overlooked problem in vision-based autonomous driving: the learner-expert asymmetry that arises when expert driving intent cannot be deduced from the learner's observation space. To tackle this, we introduce LEAD, a dataset that minimizes these asymmetries through careful collection techniques, such as constraining the expert's field-of-view. When training on LEAD, our simple TFv6 architecture surpasses all prior work on all public CARLA benchmarks by a large margin. Beyond simulation, co-training on LEAD and real-world data improves performance on the NAVSIM and Waymo Open Datasets. Our work highlights that careful dataset design, i.e., ensuring proper intent representation and realistic visibility constraints, is as crucial as algorithmic innovation for end-to-end autonomous driving.

Limitations: We identify several recurring failure modes of our driving policy. In CARLA, the model struggles to recover from route deviations. The agent frequently misses highway exits that require multiple lane changes at high speeds. While such behavior might not be desired, we find the CARLA metric is overly strict and promotes unsafe lane-change maneuvers. Finally, we observe in all tested benchmarks that navigation in dense urban environments remains challenging.

References

[1] Jens Beißwenger. Pdm-lite: A rule-based planner for carla leaderboard 2.0. URL: https://github.com/OpenDriveLab/DriveLM/blob/DriveLM-CARLA/pdm_lite/docs/report.pdf, 2024. 3

[2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv.org*, 1604.07316, 2016. 3

[3] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *Proc. Conf. on Robot Learning (CoRL)*, 2025. 3, 7, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 7

[5] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[6] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Proc. Conf. on Robot Learning (CoRL)*, 2019. 2, 3

[7] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 3

[8] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2024. 3, 4

[9] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 3

[10] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2023. 3, 4, 7, 8

[11] Kashyap Chitta, Daniel Dauner, and Andreas Geiger. Sledge: Synthesizing driving environments with generative models and rule-based traffic. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 7

[12] Felipe Codevilla, Antonio M. Lopez, Vladlen Koltun, and Alexey Dosovitskiy. On offline evaluation of vision-based driving models. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 3

[13] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2018. 3, 4

[14] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 3

[15] Marco Cusumano-Towner, David Hafner, Alex Hertzberg, Brody Huval, Aleksei Petrenko, Eugene Vinitsky, Erik Wijmans, Taylor Killian, Stuart Bowers, Ozan Sener, Philipp Krähenbühl, and Vladlen Koltun. Robust autonomy emerges from self-play. In *Proc. of the International Conf. on Machine learning (ICML)*, 2025. 3

[16] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Proc. Conf. on Robot Learning (CoRL)*, 2023. 3

[17] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3, 7, 8

[18] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proc. Conf. on Robot Learning (CoRL)*, 2017. 2, 3, 6

[19] Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. RAP: 3d rasterization augmented end-to-end planning. *arXiv.org*, 2510.04333, 2025. 8

[20] Simon Gerstenecker, Andreas Geiger, and Katrin Renz. Plant 2.0: Exposing biases and structural flaws in closed-loop driving. *arXiv.org*, 2025. 4

[21] Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, John D. Co-Reyes, Rishabh Agarwal, Rebecca Roelofs, Yao Lu, Nico Montali, Paul Mougin, Zoey Yang, Brandyn White, Aleksandra Faust, Rowan McAllister, Dragomir Anguelov, and Benjamin Sapp. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 7

[22] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, et al. Planning-oriented autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 6, 7

[23] Bernhard Jaeger. Expert drivers for autonomous driving. Master's thesis, University of Tübingen, 2021. 3

[24] Bernhard Jaeger. Transfuser versions. URL: https://github.com/autonomousvision/carla_garage/blob/leaderboard_2/docs/history.md, 2024. 3

[25] Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden biases of end-to-end driving models. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023. 3, 4, 5, 6

[26] Bernhard Jaeger, Daniel Dauner, Jens Beißwenger, Simon Gerstenecker, Kashyap Chitta, and Andreas Geiger. Carl: Learning scalable planning policies with simple rewards. *Proc. Conf. on Robot Learning (CoRL)*, 2025. 2, 3, 6

[27] 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 *Ad-*

vances in Neural Information Processing Systems (NeurIPS), 2024. 2, 4, 6

[28] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *Proc. of the International Conf. on Learning Representations (ICLR)*. OpenReview.net, 2025. 3

[29] Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, and Holger Caesar. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2024. 7

[30] Saman Kazemkhani, Aarav Pandya, Daphne Cornelisse, Brennan Shacklett, and Eugene Vinitsky. Gpudrive: Data-driven, multi-agent driving simulation at 1 million FPS. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2025. 7

[31] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 8

[32] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2022. 3

[33] Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in CARLA-V2). In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 3

[34] Tianyu Li, Yihang Qiu, Zhenhua Wu, Carl Lindström, Peng Su, Matthias Nießner, and Hongyang Li. MTGS: Multi-traversal gaussian splatting. *arXiv.org*, 2503.12552, 2025. 8

[35] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and José M. Álvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[36] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[37] Shuai Liu, Quanmin Liang, Zefeng Li, Boyang Li, and Kai Huang. Gaussianfusion: Gaussian-based multi-sensor fusion for end-to-end autonomous driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 3

[38] William Ljungbergh, Adam Tonderski, Joakim Johnander, Holger Caesar, Kalle Åström, Michael Felsberg, and Christoffer Petersson. Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 3, 7

[39] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 6

[40] Dean Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NIPS)*, 1988. 3

[41] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 4, 5, 7

[42] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3, 4, 6, 7

[43] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A controllable multi-view generative world model for autonomous driving. *arXiv.org*, 2503.20523, 2025. 7

[44] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics: Results of the 11th international conference*, pages 621–635. Springer, 2017. 3

[45] Hao Shao, Letian Wang, RuoBing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Proc. Conf. on Robot Learning (CoRL)*, 2022. 3

[46] Hao Shao, Letian Wang, Ruobing Chen, Steven L. Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[47] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Pdm-lite dataset for carla leaderboard 2.0. URL: https://huggingface.co/datasets/autonomousvision/PDM_Lite_Carla_LB2, 2024. 2

[48] 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 *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024. 2, 3, 4, 6, 7

[49] Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2025. 3, 4, 6, 7

[50] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 2000. 4

[51] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. 2024. 3

[52] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[53] Bernhard Wymann, Christos Dimitrakakisy, Andrew Sumnery, Eric Espié, and Christophe Guionneauz. Torcs: The open racing car simulator, 2015. 3

[54] Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. *arXiv.org*, 2025. 3, 7, 8

[55] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[56] 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 nuscenes. *arXiv.org*, 2023. 3

[57] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 3

[58] Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *arXiv.org*, 2024. 3, 7

[59] Julian Zimmerlin, Jens Beißwenger, Bernhard Jaeger, Andreas Geiger, and Kashyap Chitta. Hidden biases of end-to-end driving datasets. *arXiv.org*, 2412.09602, 2024. 2, 3, 5, 6, 7