NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking

Daniel Dauner[1,2] **Marcel Hallgarten**[1,5] **Tianyu Li**[3] **Xinshuo Weng**[4] **Zhiyu Huang**[4,6]
Zetong Yang[3] **Hongyang Li**[3] **Igor Gilitschenski**[7,8] **Boris Ivanovic**[4] **Marco Pavone**[4,9]
Andreas Geiger[1,2] **Kashyap Chitta**[1,2]
[1]University of Tübingen [2]Tübingen AI Center [3]OpenDriveLab at Shanghai AI Lab
[4]NVIDIA Research [5]Robert Bosch GmbH [6]Nanyang Technological University
[7]University of Toronto [8]Vector Institute [9]Stanford University

Abstract

Benchmarking vision-based driving policies is challenging. On one hand, open-loop evaluation with real data is easy, but these results do not reflect closed-loop performance. On the other, closed-loop evaluation is possible in simulation, but is hard to scale due to its significant computational demands. Further, the simulators available today exhibit a large domain gap to real data. This has resulted in an inability to draw clear conclusions from the rapidly growing body of research on end-to-end autonomous driving. In this paper, we present NAVSIM, a middle ground between these evaluation paradigms, where we use large datasets in combination with a non-reactive simulator to enable large-scale real-world benchmarking. Specifically, we gather simulation-based metrics, such as progress and time to collision, by unrolling bird's eye view abstractions of the test scenes for a short simulation horizon. Our simulation is non-reactive, *i.e.*, the evaluated policy and environment do not influence each other. As we demonstrate empirically, this decoupling allows open-loop metric computation while being better aligned with closed-loop evaluations than traditional displacement errors. NAVSIM enabled a new competition held at CVPR 2024, where 143 teams submitted 463 entries, resulting in several new insights. On a large set of challenging scenarios, we observe that simple methods with moderate compute requirements such as TransFuser can match recent large-scale end-to-end driving architectures such as UniAD. Our modular framework can potentially be extended with new datasets, data curation strategies, and metrics, and will be continually maintained to host future challenges. Our code is available at `https://github.com/autonomousvision/navsim`.

1 Introduction

Autonomous vehicles (AVs) have gained immense research interest due to their potential to change transportation and improve traffic safety [23, 9]. This has created a large community working on the development of AV algorithms, which map high-dimensional sensor data to desired vehicle control outputs. Therefore, measuring and comparing the performance of AV algorithms is a crucial task.

Unfortunately, it is extremely challenging to evaluate driving performance, and the most widely-used benchmarks today fall short in several respects: (1) the datasets used, such as nuScenes [5], were created for perception tasks such as object detection. As such, they focus on visual diversity and label quality instead of the relevance of the data for research on planning. Often, most frames have a trivial solution of extrapolating the historical driving behavior, leading to "blind" driving policies that observe only the vehicle's past trajectory obtaining state-of-the-art performance [56, 32, 16]. (2) Due to the fact that driving is an inherently multifaceted task where the algorithm must coordinate several desired properties such as safety, comfort, and progress, the evaluation metric must also balance

38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.

https://doi.org/10.52202/079017-0902

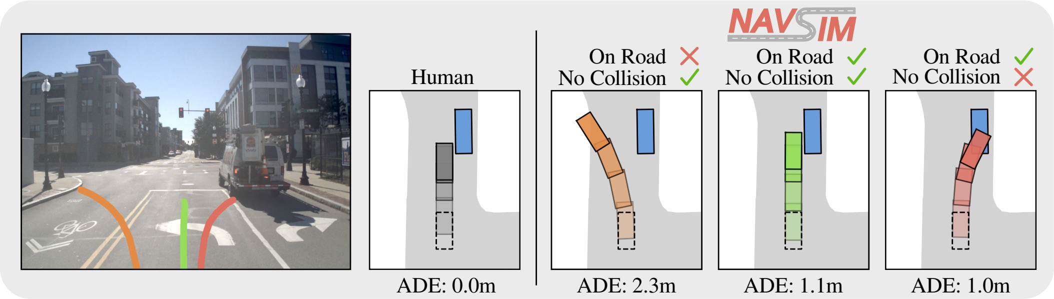

Figure 1: **NAVSIM.** Traditional metrics such as the average displacement error (ADE) overlook the multi-modality of driving. They penalize trajectories that deviate from a recorded human driving log, even if such a trajectory is safe. Our benchmark evaluates trajectory outputs of sensor-based driving policies with simulation-based metrics, considering collisions and map compliance.

potentially conflicting objectives. However, as shown in Fig. 1, existing metrics such as the average displacement error (ADE) between a predicted and recorded human trajectory often misrepresent the relative accuracy of trajectories. (3) Since driving involves interactions among multiple agents, evaluation must ideally be interactive, e.g., in simulation. Unfortunately, existing simulators with synthetic sensor data exhibit a significant domain gap to real-world driving. (4) Besides, the lack of a standardized evaluation setup has led to subtle inconsistencies between metrics in existing work, leading to unfair comparisons and inaccurate conclusions [50, 32]. Collectively, these problems hinder progress in the development of AVs, emphasizing the need for more principled benchmarks.

In this work, we take steps towards alleviating these issues. First, we propose a strategy for sampling interesting driving scenarios and apply it to the largest publicly-available driving dataset [26]. We obtain, for the first time, over 100k challenging real-world driving scenarios for training and evaluating sensor-based driving policies. We show that in these scenarios, "blind" driving policies fail to compete with more principled sensor-based policies. Second, we draw inspiration from the literature of rule-based planning for AVs [41, 18, 39, 16] to identify a set of diverse, efficient, and principled metrics that cover multiple facets of the autonomous driving task. Third, we circumvent the need for inaccurate sensor simulation with domain gaps by simplifying our simulation to a non-reactive one. Given an observed real-world sensor input, the agent under test commits to a set of actions for a specific time horizon. Further, these actions are assumed to not affect the future behavior of other agents in the scene. Under this setting, it is possible to simulate the expected motion of all agents over this time horizon in a simplified bird's-eye-view (BEV) abstraction of the scene, and incorporate metrics that involve interactions, as we observe in Fig. 1. Empirically, we demonstrate that our selected metrics are well-correlated to the outcomes of closed-loop simulations. Finally, we establish an official evaluation server on the open-source HuggingFace platform, which is free, has a low maintenance overhead, and enables future scaling to more challenging datasets and metrics.

We combine these ideas to propose NAVSIM, a comprehensive tool for AV data curation, simulation, and benchmarking. We instantiate standardized training and evaluation splits for NAVSIM with the OpenScene dataset [15], though our framework can be extended to other datasets. With these splits, we present a detailed analysis of popular end-to-end driving models previously benchmarked either exclusively on CARLA [17] or nuScenes [5], providing the first direct comparison between these families of approaches in an independent evaluation setting. Interestingly, we find that the performances of the best methods developed in both settings are similar, despite a vast difference in computational requirements for their training. Finally, we review the insights gained through the 2024 NAVSIM challenge[1], hosted in conjunction with the CVPR 2024 Workshop on Foundation Models for Autonomous Systems. For the challenge, 143 teams from 13 countries developed diverse methods that competed on the proposed benchmark. The top methods ranged from multi-billion parameter vision language models [45, 33, 53, 58] to more efficient and recently overlooked approaches based on trajectory sampling and scoring [40, 20, 10], demonstrating the remarkable ability of the broader community to advance AV research when provided with the right tools.

[1] https://opendrivelab.com/challenge2024/#end_to_end_driving_at_scale

Contributions. (1) We build NAVSIM, a framework for non-reactive AV simulation, with standardized protocols for training and testing, data curation tools ensuring broad accessibility, and an official public evaluation server used for the inaugural NAVSIM challenge. (2) We develop configurable simulation-based metrics that are well-suited for evaluating sensor-based motion planning. (3) We reimplement a collection of end-to-end approaches for NAVSIM including TransFuser, UniAD, and PARA-Drive, showcasing the surprising potential of simple models in our challenging scenarios.

2 Related Work

End-to-End Driving. End-to-end driving streamlines the entire stack from perception to planning into a single optimizable network. This eliminates the need for manually designing intermediate representations. Following pioneering work [35, 4, 27], a diverse landscape of end-to-end models has emerged. For instance, an extensive body of end-to-end approaches focuses on closed-loop simulators, utilizing single-frame cameras, LiDAR point clouds, or a combination of both for expert imitation [7, 11, 36, 8, 52, 43, 44, 12, 24, 57, 22]. More recently, developing end-to-end models on open-loop benchmarks has gained traction [20, 21, 25, 54, 32, 50]. Our work introduces a new evaluation scheme with which we compare end-to-end models from both communities.

Closed-Loop Benchmarking with Simulation. Driving simulators allow us to evaluate autonomous systems in a closed-loop manner and collect downstream driving statistics, including collision rates, traffic-rule compliance, or comfort. A broad body of research conducts evaluations in simulators, such as CARLA [17] or Metadrive [29] with sensor simulation, or nuPlan [26] and Waymax [19] for data-driven simulation. Unfortunately, ensuring realism when simulating traffic behavior or sensor data remains a challenging task. To simulate camera or LiDAR sensors, most established simulators rely on graphics-based rendering methods, leading to an inherent domain gap in terms of visual fidelity and sensor characteristics. Data-driven simulators for motion planning incorporate traffic recordings but do not support image or LiDAR-based methods [26, 19, 13]. Data-driven sensor simulation leverages and adapts real-world sensor data to create new simulations where the vehicle may move differently, but the rendering quality of existing tools is subpar [1, 2, 49]. Further, while promising image [47] or LiDAR [34] synthesis approaches exist, efficiently simulating sensors entirely from data remains an open problem. In this work, we provide an approach for the evaluation of real sensor data with simulation-based metrics by making a simplifying assumption that the agent and environment do not influence each other over a short simulation horizon. Despite this strong assumption, when benchmarking on real data, NAVSIM better reflects planning performance than established evaluation protocols, as demonstrated through our systematic experimental analysis.

Open-Loop Benchmarking with Displacement Errors. Open-loop evaluation protocols commonly measure displacement errors between trajectories of a recorded expert (i.e., of a human driver) and a motion planner. However, several issues concerning evaluation with displacement errors have surfaced recently, particularly on the nuScenes dataset [5]. Given that nuScenes does not provide standardized planning metrics, prior work relied on independent implementations, which led to inconsistencies when reporting or comparing results [50, 32]. Next, most planning models in nuScenes receive the human trajectory endpoint as a discrete direction command [20, 21, 25, 32, 50], thereby leaking ground-truth information into inputs. Moreover, about 75% of the scenarios in nuScenes involve trivial straight driving [32], leading to simple solutions when extrapolating the ego-motion. For instance, AD-MLP demonstrates that an MLP on the kinematic ego status (ignoring perception completely) can achieve state-of-the-art displacement errors [56]. Such blind agents are undeniably dangerous, which highlights a broader concern: displacement metrics are not correlated to closed-loop driving [14, 17, 3, 16]. In this work, we address prevalent issues of nuScenes and propose a standardized driving benchmark with challenging scenarios and an official evaluation server. We derive a navigation goal from the lane graph instead of the human trajectory to prevent label leakage, and propose principled simulation-based metrics as an alternative to displacement errors.

3 NAVSIM: Non-Reactive Autonomous Vehicle Simulation

NAVSIM combines the ease of use of open-loop benchmarks such as nuScenes [5] with metrics based on closed-loop simulators such as nuPlan [26]. In the following, we give a detailed introduction to the task and metrics that driving agents are challenged with in NAVSIM. Subsequently, we propose a filtering method to obtain standardized train and test splits covering challenging scenes.

Task description. Driving agents in NAVSIM must plan a trajectory, defined as a sequence of future poses, over a horizon of h seconds. Their input contains streams of *past* frames from onboard sensors, such as cameras, LiDAR, as well as the vehicle's current speed, acceleration, and navigation goal, jointly termed the ego status. For compatibility with prior work [20, 21, 25, 50], we provide the navigation goal as a one-hot vector with three categories: left, straight, or right.

Non-Reactive Simulation. Traditional closed-loop benchmarks normally infer planners at high frequencies, e.g., 10Hz [17, 26]. However, this requires efficient simulation of all input modalities for the driving agent, including high-dimensional sensor streams in the case of sensor-based approaches. To sidestep this, the core idea of NAVSIM is to evaluate driving agents using a non-reactive simulation. This means driving agents are only queried in the initial frame of each scene. Afterwards, the planned trajectory is kept fixed for the entire trajectory duration. Over this short horizon, no environmental feedback is provided to the driving agent, and the NAVSIM evaluation is purely based on the initial real-world sensor sample. This makes the agent's task more challenging, limiting simulations to short horizons. We select a horizon of $h = 4$ seconds, which has been shown in prior work to be adequate for closed-loop planning [16]. Despite this limitation, non-reactive simulation offers a key advantage: unlike traditional open-loop benchmarks, which mainly compare the planned trajectory to the human driver's trajectory in a similar setting, it enables the use of simulation outcomes to compute metrics reflecting safety, comfort, and progress. An LQR controller [28] is applied at each simulation iteration to calculate steering and acceleration values, and a kinematic bicycle model [37] propagates the ego vehicle. We execute this pipeline at 10Hz over the 4s trajectory horizon. In Sec. 4.1, we show that despite our simplifying assumption, our evaluation results in a much better alignment with closed-loop metrics than traditional open-loop metrics achieve.

PDM Score. NAVSIM scores driving agents in two steps. First, subscores in range $[0, 1]$ are computed after simulation. Second, these subscores are aggregated into the PDM Score (PDMS) $\in [0, 1]$. It is named after the Predictive Driver Model (PDM) [16], a state-of-the-art rule-based planner which uses this scoring function to evaluate trajectory proposals during closed-loop simulation in nuPlan. The metric is also an efficient reimplementation of the nuPlan closed-loop score metric [26]. In NAVSIM, the PDMS can be adapted by adding or removing subscores, changing aggregation parameters, or making subscores more challenging, e.g., by adapting their internal thresholds. It is calculated per frame and averaged across frames. In this work, we use the following aggregation of subscores:

$$\text{PDMS} = \underbrace{\left(\prod_{m\in\{\texttt{NC},\texttt{DAC}\}} \texttt{score}_m\right)}_{\text{penalties}} \times \underbrace{\left(\frac{\sum_{w\in\{\texttt{EP},\texttt{TTC},\texttt{C}\}} \texttt{weight}_w \times \texttt{score}_w}{\sum_{w\in\{\texttt{EP},\texttt{TTC},\texttt{C}\}} \texttt{weight}_w}\right)}_{\text{weighted average}}. \tag{1}$$

Subscores are categorized by their importance as penalties or terms in a weighted average. A penalty punishes inadmissible behavior such as collisions with a factor < 1. The weighted average aggregates subscores for other objectives such as progress and comfort. In the following, we briefly describe each subscore. More details can be found in the supplementary material.

Penalties. Avoiding collisions and staying on the road is imperative for motion planning as it ensures traffic rule compliance and the safety of pedestrians and road users. Thus, failing to drive with no collisions (NC) with road users (vehicles, pedestrians, and bicycles) or infractions with regard to drivable area compliance (DAC) result in hard penalties of $\texttt{score}_{\texttt{NC}} = 0$ or $\texttt{score}_{\texttt{DAC}} = 0$ respectively. This results in a PDMS of 0 for the current scene. We ignore certain collisions that are not considered "at-fault" in the non-reactive environment, e.g. when the ego vehicle is static. For collisions with static objects, we apply a softer penalty of $\texttt{score}_{\texttt{NC}} = 0.5$.

Weighted Average. The weighted average accounts for ego progress (EP), time-to-collision (TTC), and comfort (C). The ego progress subscore $\texttt{score}_{\texttt{EP}}$ represents the agent progress along the route center as a ratio to an approximated safe upper bound from the PDM-Closed planner [16]. PDM-Closed obtains a possible progress value without collisions or off-road driving with a search-based strategy based on trajectory proposals. The final ratio is clipped to $[0, 1]$ while discarding low or negative progress scores if the upper bound is below 5 meters. Next, the TTC subscore ensures that driving agents respect the safety margins to other vehicles. Defaulting to a value of 1, this subscore is set to 0 if for any simulation step within the 4s horizon, the ego-vehicle's time-to-collison, when projected forward with a constant velocity and heading, is less than a certain threshold. Finally, the comfort subscore is obtained by comparing the acceleration and jerk of the trajectory to predetermined thresholds. Following the cost weights used by the PDM-Closed planner and the 2023

https://doi.org/10.52202/079017-0902

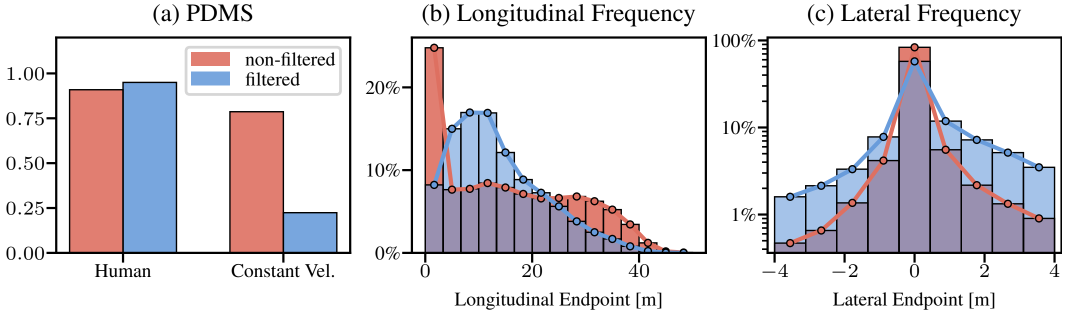

Figure 2: **Filtering.** (a) We consider challenging scenes where maintaining a constant velocity and heading fails compared to the human driver. (b) Our filtering primarily removes scenes with static or fast longitudinal movement and (c) leads to more diversity in lateral movement (log-scale).

nuPlan Challenge, we set the coefficients of the weighted average as $\texttt{weight}_{\texttt{EP}} = 5$, $\texttt{weight}_{\texttt{TTC}} = 5$, and $\texttt{weight}_{\texttt{C}} = 2$. We find this selection reasonable and robust to changes. For example, the top 3 ranks of the NAVSIM challenge remain identical when assigning an equal weight to the subscores.

3.1 Generating Standardized and Challenging Train and Test Splits

Dataset. The NAVSIM framework is agnostic to the choice of driving dataset. We choose OpenScene [15], a redistribution of nuPlan [26], the largest annotated public driving dataset. OpenScene includes 120 hours of driving at a reduced frequency of 2Hz typically considered by end-to-end planning algorithms, resulting in a 90% reduction of data storage requirements compared to nuPlan from over 20 TB to 2 TB. Our agent input, based on OpenScene, comprises eight cameras, each with a resolution of 1920×1080 pixels, and a merged LiDAR point cloud from five sensors. The input includes the current time-step and optionally 3 past frames, totaling 1.5s at 2Hz. In principle, any driving dataset that provides annotated HD maps, object bounding boxes, and sensor data can be converted into this format and thus be used with NAVSIM.

Filtering for challenging scenes. A majority of human driving data involves trivial situations such as being stationary or straight driving at a near constant speed. These can be solved efficiently by simple heuristics, e.g., as depicted in Fig. 2 (a), the baseline of maintaining a constant velocity and heading achieves a PDMS of 79% on the OpenScene dataset, where human-level performance corresponds to 91%. In NAVSIM, we propose the use of a filtered dataset to remove frames with (1) near-trivial solutions and (2) significant annotation errors. We remove highly simplistic scenes by detecting if the previously mentioned constant velocity agent exceeds a PDMS of 0.8. Similarly, we remove scenes in which the human trajectory results in a PDMS of less than 0.8. This ensures that an acceptable solution exists to these difficult scenarios and filters out noisy annotations such as inaccurate bounding boxes. These thresholds can be adjusted based on the desired filtered dataset size. The resulting scenarios are challenging, which is underlined by the score of the constant velocity agent dropping to 22%, whereas the human expert achieves a score of 95%. The higher ratio of non-trivial scenarios, such as turning, also results in endpoints being less distant longitudinally when nonzero, and more evenly distributed laterally, as seen in Fig. 2 (b-c). We employ this filtering strategy to provide standardized splits for training and testing, called `navtrain` and `navtest`, with 103k and 12k samples respectively. This curated data serves as a benchmark accessible as a standalone download option with a moderate storage demand given its large scale and diversity (450 GB).

4 Experiments

In this section, we present the results of our experiments aimed at answering the following questions: (1) Can non-reactive open-loop simulation provide sufficient correlation to closed-loop metrics? (2) What new conclusions do experiments on NAVSIM provide compared to prior benchmarks?

https://doi.org/10.52202/079017-0902

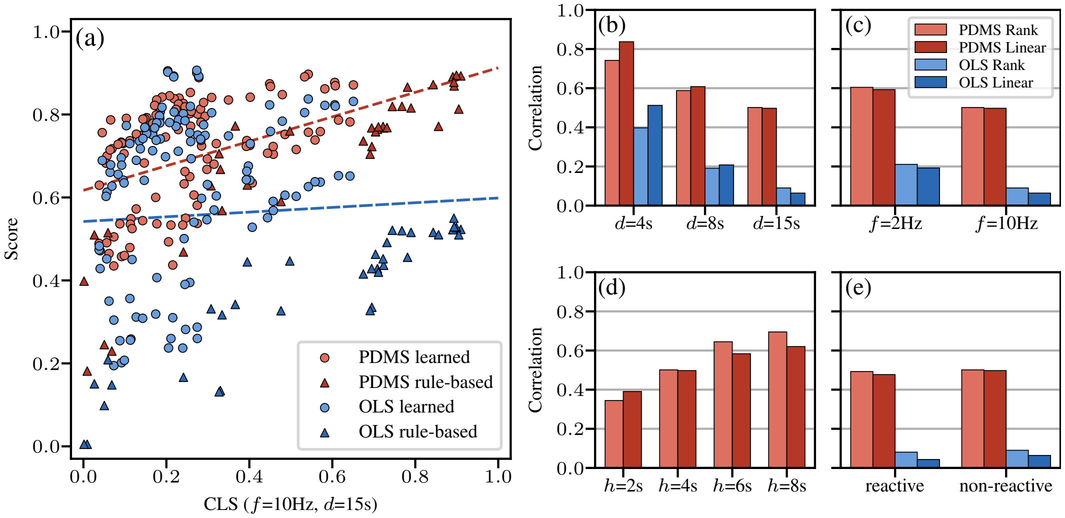

Figure 3: **Closed-Loop Alignment.** (a) For each planner, we show open-loop metrics (OLS, PDMS) together with the corresponding closed-loop score (CLS). The trendlines depicting correlations are fit linearly to all (learned and rule-based) planners. Moreover, we analyze different (b) CLS durations d, (c) planning frequencies f, (d) PDMS horizons h, and (e) closed-loop background agent behaviors.

4.1 Alignment Between Open-Loop and Closed-Loop Evaluation

Open-loop metrics should ideally be aligned with closed-loop metrics in their evaluation of different driving algorithms. In this section, we benchmark a large set of planners to analyze the alignment of closed-loop metrics with traditional distance-based open-loop metrics and the proposed PDMS.

Benchmark. Studying the relation of closed-loop and open-loop metrics necessitates access to a fully reactive simulator. To stay compatible with the dataset, we use the nuPlan simulator [26], which enables simulation for privileged planners with access to ground-truth perception and HD map inputs. Similar to PDMS, nuPlan combines weighted averages and multiplied penalties in two official scores: the **open-loop score (OLS)** aggregates displacement and heading errors with a multiplied miss-rate, and the **closed-loop score (CLS)** implements similar metrics from Section 3. Including PDMS, all metrics are in $[0, 1]$ with higher scores indicating better performance.

Due to the heavy computational requirements of closed-loop simulation, we evaluate on the `navmini` split. This is a new split we create for rapid testing, with 396 scenarios in total that are independent of both `navtrain` and `navtest` but filtered using the same strategy (Section 3.1) and hence similarly distributed. We note that nuPlan offers two kinds of background agents: reactive agents along lane centers based on the Intelligent Driver Model (IDM) [48], and non-reactive agents replayed from the dataset, which we employ unless otherwise stated. While reactive simulations of longer or dynamic lengths are generally desirable, e.g. to evaluate long-term decisions, enabling this requires dedicated solutions to long-horizon simulation that are not currently available in nuPlan [13]. Therefore, we default to a fixed closed-loop simulation duration of $d = 15$s, and a planning frequency of $f = 10$Hz, which are the standard closed-loop simulation settings in nuPlan [26].

Motion Planners. Open-loop metrics favor learned planners while rule-based approaches perform well in closed-loop evaluation in nuPlan [16]. We use a combination of both planner types in this experiment to cover different performance levels. In total, we include 37 rule-based planners with 2 constant velocity and 8 constant acceleration models, 15 IDM planners [48], and 12 PDM-Closed variants [16] which differ in hyperparameters for trajectory generation. For learned planning, we evaluate Urban Driver models [42] of 2 model sizes and 2 training lengths, and PlanCNN [38] models with 15 input combinations of the BEV raster, ego status, centerline, and navigation goal. We train all models on $\{25\%, 50\%, 100\%\}$ of `navtrain` and an equally sized uniformly sampled subset of OpenScene, giving 114 learned planners. See the supplementary material for additional details.

Results. The alignment between metrics is presented in Fig. 3 (a-e). Compared to OLS, we consistently observe better closed-loop correlation for PDMS, in terms of Spearman's (rank) and Pearson's (linear) correlation coefficients. As shown in (a), PDMS can capture the closed-loop

https://doi.org/10.52202/079017-0902

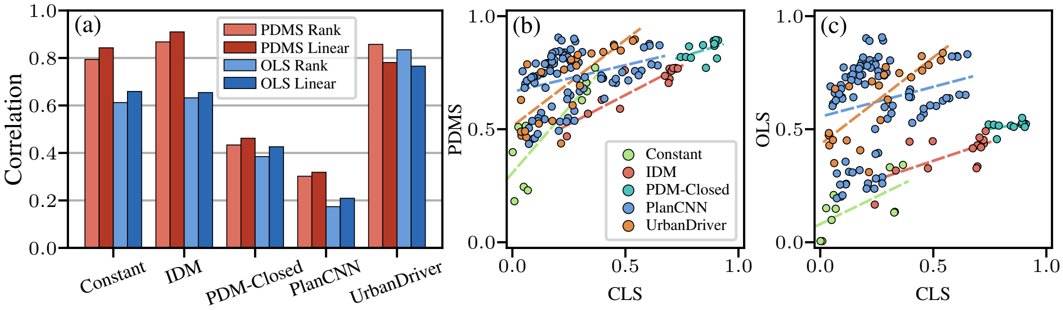

Figure 4: **Planner-Level Alignment of Metrics.** We report the correlation coefficients between open-loop metrics (OLS, PDMS) and the closed-loop score (CLS) for the five planner types considered in our study. The PDMS is better correlated to the CLS for every planner type.

Method	**Ego Stat.**	**Image**	**LiDAR**	**Video**	**NC ↑**	**DAC ↑**	**TTC ↑**	**Comf. ↑**	**EP ↑**	**PDMS ↑**
Constant Velocity	✓				68.0	57.8	50.0	100	19.4	20.6
Ego Status MLP	✓				93.0	77.3	83.6	100	62.8	65.6
LTF [12]	✓	✓			97.4	**92.8**	92.4	100	79.0	83.8
TransFuser [12]	✓	✓	✓		97.7	**92.8**	92.8	100	79.2	**84.0**
UniAD [21]	✓	✓		✓	97.8	91.9	92.9	100	78.8	83.4
PARA-Drive [50]	✓	✓		✓	**97.9**	92.4	**93.0**	99.8	**79.3**	**84.0**
Human					*100*	*100*	*100*	*99.9*	*87.5*	*94.8*

Table 1: **Navtest Benchmark.** We show the no at-fault collision (NC), drivable area compliance (DAC), time-to-collision (TTC), comfort (Comf.), and ego progress (EP) subscores, and the PDM Score (PDMS), as percentages. Relying on the ego status is insufficient for competitive results. While sensor agents improve, the gap to human performance highlights our benchmark's challenges.

properties of both learned and rule-based planners, whereas distance-based open-loop metrics show a clear misalignment. Decreasing the CLS duration in (b) from $d = 15$s to $d = 4$s further raises the correlation of PDMS and OLS, as the simulation horizon more closely matches the open-loop counterparts. Interestingly, we observe a higher correlation of open-loop metrics in (c) when reducing the planning frequency to 2Hz. We expect a lower planning frequency to mitigate cumulative errors and enhance the controller's stability in simulation, leading to more precise trajectory execution. Moreover, we observe an increase in correlation for longer PDMS horizons in (d), ranging from $h = 2$s to $h = 8$s. While predicting the future motion over 8s is challenging in uncertain scenarios, our results indicate the value of long horizons when evaluating motion planners. Lastly, replacing the non-reactive background agents with reactive IDM vehicles during closed-loop simulation in (e) has little effect on the correlation, possibly due to the similar difficulty of both tasks [16].

The imbalanced distribution of different types of planners in our study may introduce biases into the overall correlations presented in Fig. 3. To address this, we visualize the individual correlations of each planner type in Fig. 4. The correlation values vary depending on metric range and variance of each planner type. Nevertheless, when examining each type individually, the PDMS is better correlated to the CLS than the OLS, and is always positively correlated.

4.2 Analysis of the State of the Art in End-to-End Autonomous Driving

In this section, we benchmark a collection of end-to-end architectures, which previously achieved state-of-the-art performance on existing open- or closed-loop benchmarks.

Methods. As a lower bound, we consider the **(1) Constant Velocity** baseline detailed in Section 3.1. We include an **(2) Ego Status MLP** as a second "blind" agent, which leverages an MLP for trajectory prediction given only the ego velocity, acceleration and navigation goal. As an established architecture on CARLA, we evaluate our reimplementation of **(3) TransFuser** [12], which uses three cropped and downscaled forward-facing cameras, concatenated into a 1024×256 image, and a rasterized BEV LiDAR input for predicting waypoints. It performs 3D object detection and BEV semantic segmentation as auxiliary tasks. We then consider **(4) Latent TransFuser (LTF)** [12], which shares

Config	Parameter	Setting	NC ↑	DAC ↑	TTC ↑	Comf. ↑	EP ↑	PDMS ↑
A1	Default config	Seed 1	**98.0**	91.3	**94.2**	100	78.1	83.3
A2		Seed 2	97.7	92.8	92.8	100	79.2	84.0
A3		Seed 3	97.9	**93.0**	93.1	100	**79.3**	**84.4**
B1	Ego status	Goal only	96.8	91.9	91.3	98.6	77.3	81.8
B2		Goal and velocity only	96.7	92.3	91.0	100	77.8	82.3
C1	Camera FOV	60° (1 camera)	96.7	90.2	90.9	100	75.8	80.3
C2		160° (3 cameras)	97.6	91.4	92.7	100	78.1	82.8
C3		240° (5 cameras)	97.8	92.5	93.0	100	79.2	84.1
D1	LiDAR range	F:16, B:16, L:16, R:16	96.9	88.3	91.2	100	74.6	79.1
D2		F:64, B:32, L:32, R:32	97.8	92.7	93.4	100	**79.3**	84.3
D3		F:64, B:64, L:64, R:64	96.8	90.3	91.5	100	76.5	81.0
E1	Supervision	No BEV segmentation	97.4	90.5	92.2	100	77.1	81.6
E2		No 3D detection	97.8	92.7	92.9	100	79.2	84.0

Table 2: **TransFuser Ablations.** The default configuration, which obtains the best results, uses the navigation goal, velocity, and acceleration as ego status inputs. Its camera FOV is around 140° and LiDAR range is 32m to the front (F), back (B), left (L), and right (R). It uses both auxiliary tasks.

the same architecture as TransFuser but replaces the LiDAR input with a learned embedding, hence requiring only camera inputs. Moreover, we provide two state-of-the-art end-to-end architectures for open-loop trajectory prediction on nuScenes. **(5) UniAD** [21] incorporates a wide range of tasks, such as mapping, tracking, motion, and occupancy prediction in a semi-sequential architecture, which processes feature representations through several transformer decoders culminating in a trajectory planning module. **(6) PARA-Drive** [50] uses the same auxiliary tasks, but parallelizes the network architecture, such that the auxiliary task heads are trained in parallel with a shared encoder. Both UniAD and PARA-Drive use a BEVFormer backbone [31], which encodes the eight surround-view 1920×1080 camera images over four temporal frames into a BEV feature representation. Implementation details for all methods are provided in the supplementary material.

Results. We show our results on `navtest` in Table 1. The Constant Velocity model is a lower bound, as the agent is used to identify trivial driving scenes excluded from the benchmark. The Ego Status MLP achieves a PDMS of 65.6, showing the value of the acceleration and navigation goal for avoiding collisions and driving off-road. However, we observe a clear gap between agents relying solely on the ego status and those considering sensor data, in contrast to results on nuScenes [32]. All sensor agents achieve a PDMS of over 83, where TransFuser and PARA-Drive marginally perform best, with a PDMS of 84.0. Surprisingly, the camera-only LTF achieves similar results (83.8). UniAD reaches a PDMS of 83.4, which, together with PARA-Drive, do not surpass the performance of TransFuser and LTF, despite the need for more demanding training, e.g., 80 GPUs for 3 days to train PARA-Drive versus 1 GPU for 1 day for TransFuser on the `navtrain` split. Due to the definition of at-fault collisions, which discard certain rear-collisions into the ego vehicle, we suspect that surround-view cameras used by UniAD and PARA-Drive, and LiDAR input of TransFuser, are less important than the wide-angle front camera which is the only input of LTF. The 10 PDMS discrepancy to the human operator demonstrates that `navtest` poses challenges even to well-studied end-to-end architectures. Specifically, the drivable area compliance (DAC) and ego progress (EP) subscores remain the most challenging. Notably, EP cannot be solved purely by human imitation, given that the maximum progress estimate used for normalization is based on a privileged rule-based motion planner. Interestingly, all agents achieve near-perfect comfort scores, indicating that smooth acceleration and jerk profiles are learned naturally from human imitation.

Analyzing TransFuser. In Table 2, we compare several training settings for TransFuser. For the three training seeds in configs A1-A3, we observe a standard deviation of $\pm$ 0.56 in PDMS, which is relatively small compared to variance among training seeds for closed-loop simulations in CARLA [17]. Further, unlike CARLA, NAVSIM is deterministic, and we obtain identical scores when repeating evaluations of a deterministic driving agent. Discarding velocity and acceleration (B1) lowers PDMS by $1.5 - 2.6$, whereas only removing the acceleration (B2) lowers the score by $1.0 - 2.1$. We conclude that while TransFuser benefits from the ego status, it is not purely relying on the kinematic state for planning. Next, only considering the front camera (C1) with a 60° FOV leads to a small drop in almost all subscores, compared to our default setting of three cropped and concatenated images with a FOV of 140°. However, expanding the FOV with additional cameras does not result in substantially improved scores. Interestingly, restricting the LiDAR range to 16m in all directions (D1), results in a score of 79, which is lower than dropping LiDAR altogether (see LTF in Table 1). Expanding the LiDAR range to 64m in the forward direction (D2) or all directions (D3) does not provide significant improvements. We suspect that changes in the LiDAR range overly simplify or complicate the auxiliary 3D object detection and BEV semantic segmentation tasks, which operate in the LiDAR coordinate frame, hindering effective imitation learning. We check the impact of the auxiliary tasks by excluding them, where performance drops without BEV Segmentation (E1).

CVPR 2024 NAVSIM Challenge. We organized the inaugural NAVSIM challenge which ran from March - May 2024. To ensure integrity, we used a private dataset and only gave participants access to sensor inputs, withholding all annotations. Competitors could submit their agent's trajectories to our leaderboard, where they were simulated and scored to obtain the PDMS. We received 463 submissions from 143 teams, of which 78 submissions were made publicly visible. We summarize their scores in Fig. 5, relative to the constant velocity and TransFuser baselines from Table 1. The winning entry extended TransFuser and learned to predict proxy subscores for trajectory samples [30], with a sampling strategy inspired by VADv2 [10]. These predicted subscores were weighted alongside a human imitation score to select the output plan. While the idea of sampling and scoring trajectories is well-known [46, 51, 55, 6, 16], it has recently been overlooked in favor of approaches which predict a single trajectory. This result prompts a reassessment of such methods. The team that placed second employed a vision language model (VLM) for driving, which is rapidly emerging as a sub-field in the AV literature [45, 33, 53, 58]. Several submissions attempted to reimplement or extend prior work on nuScenes such as UniAD [21] and VAD [25], but were unable to outperform the TransFuser baseline by the challenge submission deadline, given the significant engineering challenge and compute requirements. The diversity of the solutions on the leaderboard shows the potential of NAVSIM as a framework for pushing the frontiers of autonomous driving research. We aim to hold future competitions with more challenging data and metrics. Detailed competition results and statistics are provided in the supplementary material.

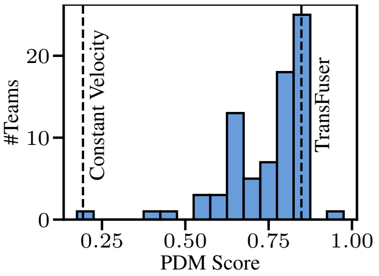

Figure 5: **NAVSIM Challenge.**

NAVSIM 1.1 Leaderboard. Due to the lasting interest after the challenge, we re-opened a public evaluation server using `navtest` as the evaluation split. The leaderboard encourages multi-seed submissions and includes reproducibility requirements for openly releasing code and model weights. We populated the leaderboard with 3 training seeds of our learned baselines, as shown in Table 3. For reference, we also include a single seed of the 2024 challenge winner [30] and constant velocity baseline. Further information is provided in the supplementary material and leaderboard webpage[2].

Method	PDMS ↑
TransFuser [12]	83.9 ± 0.4
LTF [12]	83.5 ± 0.6
Ego Status MLP	66.4 ± 0.9
Hydra-MDP [30]	91.3
Constant Velocity	20.6

Table 3: **Leaderboard 1.1.**

5 Discussion

We present NAVSIM, a framework for non-reactive AV simulation. We address shortcomings of existing driving benchmarks and propose standardized but configurable simulation-based metrics for benchmarking driving policies. For accessibility, we provide challenging scenario splits and simple data curation methods. We show that our evaluation protocol is better aligned to closed-loop driving, benchmark an established set of end-to-end planning baselines from CARLA and nuScenes, and present the results of our inaugural competition. We hope that NAVSIM can serve as an accessible toolkit for AV researchers that bridges the gap between simulated and real-world driving.

Need for Reactive Simulation. While we show improvements over displacement errors, several aspects of driving remain unaddressed by evaluation in NAVSIM. A high PDMS does not always imply a high CLS, since our framework does not consider reactiveness or the compounding accumulation

[2] https://huggingface.co/spaces/AGC2024-P/e2e-driving-navsim

of errors in closed-loop simulation. Moreover, as in CLS, rear-end collisions into the ego vehicle are currently not classified as "at-fault", resulting in little importance given to the scene behind the vehicle in NAVSIM. In the future, data-driven sensor or traffic simulation could alleviate these issues, once such methods mature and become computationally tractable. Given these limitations of the current framework, we strongly encourage the use of graphics-based closed-loop simulators, such as CARLA [17], as complementary benchmarks to NAVSIM when developing planning algorithms.

Simplicity of Metrics. As a starting point, NAVSIM offers both interpretable open-loop subscores and a scalarizing function, which lets us provide a final score and ranking for participants in the challenge. In the future, multi-objective evaluation and other aggregation functions might be required. Moreover, closed-loop metrics also face problems, i.e., PDMS inherits several weaknesses of nuPlan's CLS. Both scores do not regard certain traffic rules (e.g., stop-sign or traffic light compliance) or concepts such as transit and fuel efficiency. In the future, we aim to improve the subscore definitions (e.g. the at-fault collision logic) and add more subscores during aggregation.

Call for Datasets. Certain limitations of the nuPlan dataset persist in NAVSIM, such as missing classes in the label space, minor errors in camera parameters, or noise in vehicle poses and 3D annotations. Our analysis might favor methods that are robust to such inconsistencies. In addition, the lack of road elevation data in our representation presents a challenge for integrating scenarios based on 3D map annotations. We aim to support more datasets in the future, and advocate for more open dataset releases by the community for accelerating progress in autonomous driving.

Acknowledgments

This work was supported by the ERC Starting Grant LEGO-3D (850533), the DFG EXC number 2064/1 - project number 390727645, the German Federal Ministry of Education and Research: Tübingen AI Center, FKZ: 01IS18039A and the German Federal Ministry for Economic Affairs and Climate Action within the project NXT GEN AI METHODS. We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Daniel Dauner and Kashyap Chitta. We also thank HuggingFace for hosting our evaluation servers, the team members of OpenDriveLab for their organizational support, as well as Napat Karnchanachari and his team from Motional for open-sourcing their dataset and providing us the private test split used in the 2024 NAVSIM Challenge.

References

[1] Alexander Amini, Igor Gilitschenski, Jacob Phillips, Julia Moseyko, Rohan Banerjee, Sertac Karaman, and Daniela Rus. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[2] Alexander Amini, Tsun-Hsuan Wang, Igor Gilitschenski, Wilko Schwarting, Zhijian Liu, Song Han, Sertac Karaman, and Daniela Rus. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2022.

[3] Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Proc. Robotics: Science and Systems (RSS)*, 2019.

[4] 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.

[5] 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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

https://doi.org/10.52202/079017-0902

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

[8] 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.

[9] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv.org*, 2306.16927, 2023.

[10] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv.org*, 2402.13243, 2024.

[11] 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.

[12] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2023.

[13] 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.

[14] 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.

[15] OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. https://github.com/OpenDriveLab/OpenScene, 2023.

[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.

[17] 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.

[18] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo EM motion planner. *arXiv.org*, 1807.08048, 2018.

[19] 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.

[20] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[21] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[22] 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.

[23] Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. *Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art*, volume 12. Foundations and Trends in Computer Graphics and Vision, 2020.

https://doi.org/10.52202/079017-0902

[24] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[25] 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 *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023.

[26] 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.

[27] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John Mark Allen, Vinh Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. *arXiv.org*, abs/1807.00412, 2018.

[28] Norman Lehtomaki, Nils Sandell, and Michael Athans. Robustness results in linear-quadratic gaussian based multivariable control designs. *IEEE Trans. on Automatic Control (TAC)*, 1981.

[29] 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)*, 45(3):3461–3475, 2022.

[30] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and Jose M. Alvarez. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv.org*, 2024.

[31] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[32] 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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[33] Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. *arXiv.org*, 2312.00438, 2023.

[34] Sivabalan Manivasagam, Ioan Andrei Bârsan, Jingkang Wang, Ze Yang, and Raquel Urtasun. Towards zero domain gap: A comprehensive study of realistic lidar simulation for autonomy testing. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 8272–8282, 2023.

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

[36] 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.

[37] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[38] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, Sophia Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. In *Proc. Conf. on Robot Learning (CoRL)*, 2022.

[39] Abbas Sadat, Mengye Ren, Andrei Pokrovsky, Yen-Chen Lin, Ersin Yumer, and Raquel Urtasun. Jointly learnable behavior and trajectory planning for self-driving vehicles. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[40] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[41] Axel Sauer, Nikolay Savinov, and Andreas Geiger. Conditional affordance learning for driving in urban environments. In *Proc. Conf. on Robot Learning (CoRL)*, 2018.

[42] Oliver Scheel, Luca Bergamini, Maciej Wolczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Proc. Conf. on Robot Learning (CoRL)*, 2021.

[43] 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.

[44] 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.

[45] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with graph visual question answering. *arXiv.org*, 2312.14150, 2023.

[46] Sebastian Thrun, Michael Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia M. Oakley, Mark Palatucci, Vaughan R. Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary R. Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara V. Nefian, and Pamela Mahoney. Stanley: The robot that won the DARPA grand challenge. *Journal of Field Robotics (JFR)*, 23(9):661–692, 2006.

[47] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. NeuRAD: Neural rendering for autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

[49] Tsun-Hsuan Wang, Alexander Amini, Wilko Schwarting, Igor Gilitschenski, Sertac Karaman, and Daniela Rus. Learning interactive driving policies via data-driven simulation. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2022.

[50] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[51] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2010.

[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.

[53] Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. LLM4Drive: A survey of large language models for autonomous driving. *arXiv.org*, 2311.01043, 2023.

[54] Tengju Ye, Wei Jing, Chunyong Hu, Shikun Huang, Lingping Gao, Fangzhen Li, Jingke Wang, Ke Guo, Wencong Xiao, Weibo Mao, et al. Fusionad: Multi-modality fusion for prediction and planning tasks of autonomous driving. *arXiv.org*, 2023.

[55] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

https://doi.org/10.52202/079017-0902

[56] 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.org*, 2305.10430, 2023.

[57] Jimuyang Zhang, Zanming Huang, and Eshed Ohn-Bar. Coaching a teachable student. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[58] Yunsong Zhou, Linyan Huang, Qingwen Bu, Jia Zeng, Tianyu Li, Hang Qiu, Hongzi Zhu, Minyi Guo, Yu Qiao, and Hongyang Li. Embodied understanding of driving scenarios. *arXiv.org*, 2403.04593, 2024.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See supplementary pdf.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Abstract & supplementary pdf.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 & supplementary pdf.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplementary pdf.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See supplementary pdf.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Abstract, code as URL.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See supplementary pdf.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See supplementary pdf.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No use of crowdsourcing or research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No use of crowdsourcing or research with human subjects.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No use of crowdsourcing or research with human subjects.

https://doi.org/10.52202/079017-0902